Improving Exploration of Tactile Graphics by Visually Impaired People: Theoretical Advances and a Novel Mobile Application

Maralbek Zeinullin

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Engineering
College of Science and Engineering
University of Glasgow

September 2024

Abstract

The goal of this PhD study is to improve the accessibility of tactile graphics for people with visual impairments. Tactile graphics provide a means for blind individuals to understand non-textual information through touch. However, they often require supplementary audio or Braille text descriptions, which can be time-consuming and create obstacles to learning. For instance, some individuals may find that reading Braille or listening to a screen reader takes longer because they have to go through all the text or audio to locate the necessary information. To address these challenges, this study develops and evaluates TAURIS (Tactile AUdio Responsive Intelligent System), a novel system that provides real-time audio descriptions that accompany tactile graphics. TAURIS is comprised of pre-labelled tactile graphics, an interactive web tool for labelling, and a mobile application. The mobile application relies on a customised deep learning computer vision model to detect the user's fingertips as they explore the tactile graphics and provides information about what they are touching. Notably, no system simultaneously detecting the fingertips of both hands in real-time and under different lighting conditions has been implemented on a mobile device.

In order to evaluate the efficacy of TAURIS, a mixed-methods approach was employed. This approach consisted of structured interviews, experiments, and post-experimental feedback sessions involving a total of 20 participants, including one pilot participant. The structured interviews were designed to gather information on the participants' experiences with tactile graphics and mobile devices. In the subsequent experiments, response times and the number of correct answers were recorded during end-user testing of the app. The resulting data was analysed using the Wilcoxon signed-rank test. The analysis revealed three statistically significant findings: TAURIS allowed for faster interaction with tactile graphics and higher accuracy in answering questions compared to both Braille text and screen readers. Additionally, while no significant difference in memory retention was observed between TAURIS and Braille text, TAURIS demonstrated a significant advantage in memory retention over screen readers. Additionally, the participants provided feedback on their impressions, comments, and suggestions for improving the system. Finally, a mixed-methods approach was used to triangulate the data from multiple sources and strengthen the validity of these findings.

Based on these findings, TAURIS has the potential to empower individuals with visual impairments by providing an accessible and efficient tool that supports independent learning and

improves knowledge retention. Further research with a larger number of participants in various schools for the blind and countries would be valuable for gaining additional insights, increasing the power of the statistical analysis, and enabling further comparisons.

Contents

List of Tables

List of Figures

Acronyms

AI Artificial Intelligence

AT Assistive Technology

BANA Braille Authority of North America

CNN Convolutional Neural Network

COCO Common Objects in Context

CPU Central Processing Unit

CV Computer Vision

DL Deep Learning

DNN Deep Neural Network

FPN Feature Pyramid Network

GPU Graphical Processing Unit

GUI Graphical User Interface

HCI Human-Computer Interaction

HOG histogram of oriented gradients

IoU Intersection over Union

LBP local binary patterns

mAP Mean Average Precision

O&M orientation & mobility

OCR Optical Character Recognition

R-CNN Region-based Convolutional Neural Network

RPN Region Proposal Network

SSD Single-shot Detector

STEM science technology engineering and mathematics

SVG Scalable Vector Graphics

TG Tactile Graphics

TGH Tactile Graphics Helper

TPU Tensor Processing Unit

TTS Text-To-Speech

VIP Visually Impaired People

VR Virtual Reality

YOLO You Only Look Once

Acknowledgements

I would like to take this opportunity to express my sincere gratitude to all those who have supported me throughout my PhD journey.

First and foremost, I would like to thank my supervisor Dr. Marion Hersh for her invaluable guidance, support, and encouragement throughout my research. I am deeply grateful for her insightful feedback, unwavering support, and patience during the many challenges that I faced during my PhD. I will truly miss our conversations and the wisdom she imparted to me.

I am deeply thankful to my examiners, Dr. Nicolas Hine and Dr. Stephen Brewster, for their time, effort, and valuable feedback during the examination process. Their insights and expertise have greatly contributed to the refinement and finalisation of my thesis.

I also want to express my appreciation to my granddad, who instilled in me a love of learning from a young age. He encouraged me to study hard and pursue my academic goals. His words of wisdom have always been a source of inspiration to me. This thesis is dedicated to his memory, as he passed away during my PhD journey. His legacy of kindness and perseverance continues to inspire me every day.

I would also like to extend my deepest gratitude to my parents, who have always believed in me and supported me in every step of my academic journey. Their unwavering faith and encouragement have been a driving force behind my success, and I am forever grateful for their love and support. Their constant support and guidance have helped me to achieve my goals and have been a source of inspiration for me. Thank you, Mom and Dad, for everything you have done for me. I am honoured to have you as my parents.

I am also deeply indebted to my wife, Shakhrizada, for her unwavering support, encouragement, and sacrifices she made throughout my PhD journey. Her love and encouragement have been a constant source of motivation for me, and I could not have done this without her.

I would like to express my heartfelt gratitude to my two daughters, Aisha and Amina, who brought so much joy to my life during my PhD journey. They were my source of relief during stressful times, and their love have been a constant source of strength for me.

Finally, I am deeply grateful to the Bolashak scholarship program, which has provided me with financial support to pursue my doctoral studies. This scholarship has been instrumental in enabling me to achieve my academic aspirations, and I recognise the significant investment that the government has made in my future. I am also grateful to all the taxpayers of Kazakhstan

whose contributions have made this scholarship program possible

Declaration

I, hereby declare that the work presented in this thesis is entirely my own original work, unless otherwise indicated and appropriately cited. The research work contained in this thesis was carried out under the supervision of Dr. Marion Hersh at the University of Glasgow and in accordance with the regulations and guidelines of the University.

Furthermore, I would like to state that some of the results presented in this thesis have been published in the following peer-reviewed journal:

- **Maralbek Zeinullin** and Marion Hersh. "Tactile Audio Responsive Intelligent System." IEEE Access 10 (2022): 122074-122091.

Maralbek Zeinullin, September 2024

Chapter 1

Introduction

1.1 Context

Visual impairment is a significant public health concern that affects millions of people worldwide. According to the statistics there are 36 million individuals classified as blind and 217 million with moderate to severe vision problems. (Ackland, Resnikoff, and Bourne, 2017). Visual impairment can have substantial social and economic impacts, but it can also create barriers to learning and accessing information, especially in educational settings where visual materials are mostly used. These materials can be challenging for Visually Impaired People (VIP) to access and interpret and thus, can impede their academic progress and limit their future career opportunities.

While vision impairment can present challenges in daily life, it is important to focus on the ways in which technology can help overcome these obstacles and promote equal opportunities for all. In recent years, there has been a significant increase in the development and availability of Assistive Technology (AT) for the visually impaired, ranging from smartphone apps and wearable devices to advanced software programs. These technologies are designed to address a wide range of needs, such as navigating in unfamiliar environments, reading printed materials and engaging with the digital world. By acknowledging the benefits of AT and exploring the latest advances in the field, we can gain a better understanding of how technology can empower VIP and enrich their lives. This thesis introduces an AT system that improves the accessibility of educational materials for the visually impaired, thereby providing them with the equal opportunities to succeed in their academic pursuits.

1.2 Assistive Technology

For people with visual impairments, information encoded in a visual format creates certain barriers. In order to alleviate this, special AT devices and materials are utilised. Assistive technology systems have been defined as “equipment, devices and systems that can be used to overcome

the social, infrastructure and other barriers experienced by disabled people that prevent their full and equal participation in all aspects of society" (Hersh and Johnson, 2008). The authors categorised the AT into different areas based on activities that individuals engage in:

1. Mobility (orientation and navigation),
2. Communication and information
3. Cognitive activities
4. Daily living
5. **Education** and employment
6. Recreational activities

Education is one of the areas in which this technology can be particularly impactful. With the advancements in AT, including Tactile Graphics (TG), screen readers and braille displays, VIP can now access and interact with visual content more independently. In the subsequent sections of this chapter, detailed descriptions of these tools are provided.

1.2.1 Tactile Graphics

TG usage in AT allows visually impaired individuals to convey non-textual information by touch. To be more precise, TG is the raised line versions of graphical illustrations which were adapted for the tactual sense. The usage of these graphics is particularly crucial in the fields of science technology engineering and mathematics (STEM) and orientation & mobility (O&M) where most of the data is represented in diagrams, charts, figures and spatial maps. The most widely used methods for producing TG are:

1. Thermoform (vacuum formed)
2. Swell paper (printed image lines raise after heating)
3. Embossed (printed on Braille paper)
4. Handmade (using the thing from the environment i.e. leaves, spaghetti)

In this study swell paper-based graphics are utilised. The swell, also known as Minolta or capsule, paper is a cream-coloured paper which contains heat-reactive chemicals in it. When black ink is applied on its surface and then heated, the chemicals fracture and the area under the ink inflates. The standard printer can be used for the black ink placement and a special heat fuser is required for the image to swell up. Figure 1.1 illustrates the TG sample created using

this method. TG must be carefully designed to convey the intended meaning of the visual information, often requiring the use of texture, shape, and spatial relationships to represent objects and scenes. There are special rules provided by the Braille Authority of North America (BANA) to assure the standardisation of Braille and TG. Teachers and instructors of VIP should follow these guidelines when developing the TG samples.

In addition to the challenges of creating meaningful TG, VIP often require additional information to fully understand the content. Audio descriptions or braille annotations may be necessary to provide additional context and detail, such as labels, titles, or captions. These annotations can also help to clarify complex or abstract concepts that may be difficult to represent solely through TG. My research aims to enhance the accessibility of TG by integrating accompanying audio feedback.

Figure 1.1: Tactile Graphics printed on a swell paper

1.2.2 Braille text

Braille is a tactile writing system that allows visually impaired individuals to read and write through touch. It was invented by Louis Braille in 1824 (Sakula, 1998) and has since become the standard reading and writing system for the blind. In the system, letters, numbers and punctuation marks are represented by raised dots arranged in a variety of patterns (Figure 1.2). Braille literacy plays a crucial role for individuals with visual impairments as it allows them to acquire skills in spelling, grammar and punctuation. In addition to facilitating language acquisition, the Braille literacy has been linked to several positive outcomes. For instance, Ryles (1996) discovered that individuals who rely on Braille as their primary literacy mode have significantly higher employment rates, are more likely to graduate from educational institutions and achieve greater financial self-sufficiency compared to their non-Braille reading peers. There are, how-

ever, certain limitations that must be acknowledged. The primary limitation of utilising Braille text is its space requirement, which can be up to seven times larger than printed text (Johnson, 1996). In my work, I compare the efficiency of the developed mobile application against Braille text descriptions for TG. In conclusion, while Braille text has certain limitations, it remains a vital tool that not only enables individuals to read and write, but also enhances their cognitive abilities. This, in turn, is essential for achieving academic success and accessing employment opportunities.

Figure 1.2: Braille text embossed on Braille paper
Note: https://pixabay.com/photos/braille-hands-keys-read-5498805/ accessed 01/03/2023

1.2.3 Screen Readers

Screen readers are software applications that enable VIP to access digital content by converting text into synthesised speech or braille output. Optical Character Recognition (OCR) and Text-To-Speech (TTS) technologies have had a significant impact on the development and performance of screen readers. OCR allows the conversion of handwritten or printed texts into machine-readable digital ones. This technology has made it possible for screen readers to access printed and scanned text, expanding the amount of information that is available to blind individuals. TTS technology has improved the naturalness and quality of synthesized speech, making screen readers more pleasant and comfortable to use. Together, these technologies have greatly increased the accessibility of electronic devices for VIP and have transformed the way they interact with digital content.

In 1986, IBM introduced its first screen reader, which was developed to help VIP access computers and software applications (Adams et al., 1989). The IBM Screen Reader device utilised a speech synthesizer to read the text displayed on the computer screen, making it possible for visually impaired individuals to access digital information independently. The creation of this device was a pivotal breakthrough in the field of accessibility, as it was the first commercially

available screen reader to facilitate computer interaction for VIP. The technology utilised in IBM Screen Reader has since undergone significant advancement and development, and continues to be utilised by visually impaired individuals today. In addition to the Braille text descriptions, this study compares the efficiency of the developed application against screen reader descriptions for TG.

1.2.4 Common limitations of existing Assistive Technology

Number of systems were developed to make educational materials more accessible to VIP (Fusco and Morash, 2015; Baker et al., 2014; Melfi et al., 2020; Hosokawa, Miwa, and Hashimoto, 2020). Section 2.4 of the thesis reviews these and other systems in more detail. Despite these advances, there are still numerous challenges when it comes to accessing technology. There are several issues which may impact the effectiveness of AT in this context. The following examples are the most common issues associated with this technology:

- **Training:** AT can be complex and challenging to use for those who are not familiar with it. Educational institutions may not always provide sufficient training to visually impaired students, which can limit their ability to use the technology effectively.

- **Cost:** AT can be expensive, creating financial barriers for visually impaired individuals who may not have access to the necessary resources to purchase the tools they need.

- **Portability:** While many of these devices are small and compact, a significant portion of AT for the blind is not portable. For example, many braille embossers and larger TG displays are designed to be used in a stationary setting, such as an office or classroom. This can be a hindrance for those who want to take their devices on the go, whether for work or leisure.

- **Maintenance:** Like other tools, AT tools may require regular maintenance or upgrades to ensure they remain effective. If these requirements are not met, the tools may become obsolete or stop working correctly, which can impact the student's learning.

Addressing these issues requires a concerted effort from policymakers, educators and researchers to ensure that visually impaired individuals have access to the necessary tools and support to succeed in their education. This may involve providing training, ensuring compatibility between tools and software, addressing accessibility concerns, and providing financial assistance to those in need. One of the aims of this thesis is to explore these challenges and to propose a potential solution to improve accessibility.

1.3 TAURIS System

Tactile Audio Responsive Intelligent System (TAURIS) was developed to accompany TG with audio descriptions. The system consists of three components: the pre-labelled TG, an interactive labelling web tool and the phone application. The digital version of the graphics first needs to be labelled by teachers using the developed web tool. Then, the phone app, which is based on the Android platform, will accompany those graphics with the audio descriptions. The developed educational mobile application relies on a deep learning computer vision model to detect the user's fingertips. The fundamental purpose of the developed app is to allow the user to gain information without sighted assistance. Chapter 4 of the thesis presents a thorough description of the system.

1.4 Research Questions

The TAURIS app was developed with the aim of improving accessibility to TG for VIP. In order to assess the effectiveness of the app for use in education for the blind, a series of research questions were investigated through end-user interviews and testings. The purpose of this investigation was to evaluate the app's performance and usability compared to the traditional methods (Braille text and screen reader). The following questions were explored:

1. *What are visually impaired individuals' **perceptions and attitudes** toward the use of smartphone app in the context of exploring Tactile Graphics (TG)?*

2. *To what extent does real-time speech output, integrated with tactile exploration, enhance the **comprehension and retention** of complex information conveyed through TG for visually impaired users?*

3. *What methods can be employed to improve **camera aiming** in smartphone-based assistive technology applications designed for exploring TG?*

To answer these research questions, the end-user testings of the developed system were designed. Then, schools for visually impaired individuals in Glasgow were contacted. However, due to the lockdown measures in Scotland, in-person meetings were prohibited. Consequently, I decided to conduct end-user testing of the TAURIS in a school located in my home country of Kazakhstan. It is worth noting that Kazakhstan has a higher prevalence of blindness, with over 330 blind individuals per 100,000 (Atlas, 2020), as compared to approximately 150 per 100,000 in Scotland (Boswell and Kail, 2016). According to the source, there are eight schools for blind and visually impaired individuals in Kazakhstan (NNPCPK, 2022). After contacting several of them, one school agreed to participate in the research.

Following the selection of the research methodology and experiment design, ethical approval was obtained from the university. Then, I travelled to Kazakhstan to conduct the experiments

and collect data. As stated above, the initial plan to conduct the experiments in Glasgow had been cancelled, and as such, data collection was shifted to the summer period. This change in schedule resulted in a smaller number of participants taking part in the research as many students and school staff were on holiday. The data collection process was divided into three parts: the first part involved gathering general demographic information from the participants, while the second part involved conducting actual testing of the app. Finally, participants' feedback on the system was recorded. Chapter 5 of the thesis provides a more detailed description of the data collection process.

1.5 Thesis Structure

The present thesis comprises seven chapters, including the Introduction chapter. The following is an overview of each chapter:

Chapter 2 - Literature Review

This chapter provides a comprehensive review of the relevant literature in the field. Specifically, it begins by presenting a state-of-the-art of TG and the barriers that have impeded its widespread utilisation. Additionally, the chapter examines the current state of fingertip detection methods and evaluates their effectiveness in the context of TG. Finally, the chapter reviews existing educational systems for individuals with visual impairments, providing a foundation for the proposed system's development and implementation.

Chapter 3 - Fingertip Detection

This chapter introduces the fingertip detection and tracking algorithm, a core component of the TAURIS app. It details the development and implementation of a novel algorithm designed for real-time, accurate fingertip detection on TG. Furthermore, it explains the enhancements made to improve the model's accuracy and tracking stability, including strategies to handle occlusions and varying lighting conditions.

Chapter 4 - TAURIS System

This chapter provides an overview of the developed system, with a focus on the mobile application and algorithms utilised. A detailed description of these components is included. Additionally, the web tool used for annotating the TG is described in detail.

Chapter 5 - Methodology

This chapter presents the research methodology and design used in this study. The data collection methods and procedures are described in detail to ensure a comprehensive understanding

of the research process. Additionally, an overview is provided of how the collected data will be analysed.

Chapter 6 - Results and Discussion

This chapter presents the results of the experiments conducted to address research questions 1 and 2. The Likert scale questions are also analysed to shed light on research question 1. To triangulate the data and confirm the findings, the quantitative and qualitative results are combined. Additionally, the results of the after experiment interviews are analysed to investigate research question 3.

Chapter 7 - Conclusions

This chapter provides comprehensive conclusions to the study, summarising the results and their implications for each research question. The contributions of the work are discussed in detail, as well as its limitations. Finally, plans for future work are outlined to address any unanswered questions and further advance the research in this field.

Chapter 2

Literature Review

2.1 Introduction

In recent years, there has been a growing interest in developing innovative technologies that can assist Visually Impaired People (VIP) in accessing and understanding graphical information. This thesis focuses on the development of a system that combines Tactile Graphics (TG) with audio descriptions to enhance the ability to perceive and comprehend information. In this literature review chapter[1], I examine three key topics related to the development of this system.

First, I provide an overview of the significance of TG in Section 2.2. TG have been widely used as a means of representing graphical information for VIP, but their effectiveness is limited without additional information, such as verbal or audio descriptions. I review previous research on the use of TG for blind individuals and discuss the challenges that must be overcome to develop an effective system. Then, I review the key research on hand use and exploration strategies in TG. Finally, I review the camera use by blind people is Section 2.2.3.

Second, in Section 2.3 I review existing fingertip detection and tracking algorithms. This algorithm is an essential component of the developed system, as it allows VIP to interact with the TG in real-time. Finally, I discuss various educational systems that have been developed for VIP in Section 2.4. I examine the strengths and weaknesses of these solutions and identify areas where proposed system can provide additional benefits.

Overall, this section provides a comprehensive overview of the key topics related to the development of a system that combines TG with audio descriptions. Through a thorough examination of previous research and existing technologies, this review identifies gaps in the current body of knowledge and establish a foundation for the development of an innovative and effective solution.

[1]Some of the work in this chapter has appeared in Zeinullin and Hersh (2022). Maralbek Zeinullin is the first author and main contributor to this paper.

2.2 Tactile Graphics

2.2.1 Importance of Graphic Literacy and Tactile Graphics

Graphic literacy is the ability to convey information presented in the form of shapes, diagrams, maps, schemes, photos and other 2D formats. Graphics have three main advantages. They are concise, relatively easily memorable and can clearly represent relationships between data. A well drawn and labelled image can present detailed information which does not require much more than a glance to understand. Several studies have demonstrated the ability to remember visuals better than text (Paivio, 2013; Grady et al., 1998). Charts and graphs can be used to represent the links between complicated data in an easy-to-understand format (Novak and Cañas, 2008). In summary, graphics can enable the assessment and understanding of a large amount of information relatively quickly and comprehensively.

Visual graphics and TG both convey information through the use of images, but while visual graphics are designed to be seen, tactile ones are designed to be touched and felt by individuals who are blind. Numerous studies have been carried out to investigate the user experience with TG. In a survey conducted by Zebehazy and Wilton (2014a) visually impaired students (n=59) were more likely to agree (or strongly agree) that they liked using TG and wanted more access to the materials. The authors also report (Zebehazy and Wilton, 2014c) that TG help VIP to keep up with their sighted peers and to feel connected with the class teaching flow. In another research carried out by Fusco and Morash (2015) all of the participants (n=3) liked TG and pointed out their versatility in science technology engineering and mathematics (STEM) courses. Prescher, Bornschein, and Weber (2014) asked the users about their experiences with TG exploration and 56 out of 76 expressed a positive (medium to very high) attitude. However, more than half of them preferred descriptions provided along with the graphics. One of the limitations of this research is the lack of information about participants' backgrounds, including whether they attended a school for the blind or a mainstream school.

Almost in all of the surveys conducted on this research topic, participants mentioned that it is important to provide accompanying descriptions in Braille, audio or other accessible formats. The interview responses collected by Sheppard and Aldrich (2001) support the idea that TG alone are not self-sufficient especially in a secondary and high-school curriculum. Various feedback modalities which make TG content more accessible are described in the next section.

Some studies provide information about TG usefulness from the teachers and instructors perspective. Zebehazy and Wilton (2014b) surveyed more than 200 teachers of students with visual impairments in Canada and the USA. According to the results, 98% of respondents (n = 241) agreed (or strongly agreed) that exposure to TG at an early age is crucial. In further research, the authors found that students who begin learning graphics (including tactile ones) in early grades are more successful in the later academic curriculum (Zebehazy and Wilton, 2014a). In a recent study conducted by Rosenblum, Cheng, and Beal (2018), one of the teachers

reported that students with high graphic literacy skills are better at generalising and structuring information. Sheppard and Aldrich (2001) interviewed 24 teachers who worked with VIP and all of them reported that there were situations in which TG contributed substantially to effective course learning. The main limitations of these studies are the lack of information about how long the teachers have been working with VIP and how many visually impaired students they teach. To sum up, graphic literacy is very important skill, which is required for the VIP to acquire information during the education process. Many studies support the theory that TG accompanied by accessible feedback is useful learning material which helps the users to improve those skills.

2.2.2 Hand Use in Tactile Graphics Exploration

A crucial aspect of designing effective and accessible educational materials for VIP is understanding how they interact with TG. Although previous studies have often focused on recognition rates, a more detailed examination of hand movements and their specific roles during tactile exploration is required for a thorough understanding of the processes. This section summarizes key findings relating to hand dominance, functional asymmetry and the engagement of individual fingers in tactile exploration tasks.

Contrary to the notion of single-finger dominance, studies consistently reveal that VIP predominantly employ two hands during tactile exploration. Wijntjes et al. (2008b) demonstrated that participants utilized both hands in over 83% of their exploration time, a behavior that was shown to correlate with higher rates of identification. Furthermore, study by Guerreiro et al. (2015) had also shown that participants prefer two hands to explore TG. These findings highlight the significance of considering bimanual exploration in the design of TG.

While the use of two hands is the common approach, they often perform different functions. Bardot et al. (2017) have shown that VIP often use the non-dominant hand (frequently the left) to maintain contact with the graphic, acting as an "anchor" point. The dominant hand, on the other hand, engages in more refined explorations. This observation is further supported by Zhao, Kaixing, et al. (2021), who found that the right hand's tactile fixations have a significantly longer duration. This functional asymmetry indicates a sophisticated interplay between the two hands during tactile tasks.

Specifically focusing on finger use, Symmons and Richardson (2000) have observed that index fingers are the main tactile sensoring during exploration. This finding is supported by reports from participants in a study by Bahrin, Yusof, Na'im Sidek, and Ghazali (2024), who identified the index finger as the dominant one during TG exploration.

In conclusion, the available evidence points towards bimanual exploration, where each hand performs distinct roles, as the common exploration mode for VIP when using TG. The non-dominant hand tends to serve as an anchor point, while the dominant hand carries out more detailed explorations. Therefore, designers should consider this functional asymmetry when designing tactile graphics. Also, future research should aim to better understand the roles of

each hand and finger during tactile exploration.

2.2.3 Camera Use by Blind People

With the development of mobile technology, more and more visually impaired users desire to utilise phone cameras in their daily activities. However, accurate aiming of the camera remains a challenging task for them. Since proper usage of the TAURIS app requires this skill as well, it was decided to investigate this subject thoroughly. According to Jayant et al. (2011), 71% of the visually impaired respondents (out of 118) indicated that they use a phone camera regularly. The results of a survey conducted within my research showed that more than 83% (out of the 12) of participants use their phone cameras at least once a month. All of them pointed out that sometimes they experience difficulties with proper camera aiming.

In the recent decade, many researchers have attempted to solve this problem. In the early 2010s, authors such as (Bigham et al., 2010; Vázquez and Steinfeld, 2012; Balata, Mikovec, and Neoproud, 2015) used classic Computer Vision (CV) techniques to locate objects of interest. One major drawback of this approach is that the detection was not accurate. Nowadays, AI-based solutions are widely used, as they are more robust. Lee et al. (2019) designed a mobile application that provides audio-haptic feedback in real time to navigate the location of the object of interest in the frame. This was done by utilising a DNN detection model to locate the centre of the object. The authors have carried out a series of experiments with end users (N = 9). Results show that desired objects were included completely in 92% of photos taken. However, the main weakness of the study is that the proposed solution is constrained by the number of objects that the model was trained to identify. In other words, the app will not be able to detect objects the model is not familiar with.

In the same vein, Zhao, Wu, et al. (2018) in their work tried to address the issue of proper camera aiming while taking photos of people. The authors integrated Facebook's face recognition model into their bot. The developed accessibility bot is capable of providing various information including face locations, expressions, and identities. Results of their study show that in general participants found this bot helpful. However, some of them reported that this bot is suitable only for gathering with close friends and family. Another limitation is related to privacy concerns. The face recognition algorithm is being processed on the cloud and requires photos to be uploaded on a server.

A slightly different approach was used by Iwamura et al. (2020). In contrast to previous works, the proposed system generates an image after the photo was taken. First, the user captures an image using an omnidirectional camera (360°-degree camera). Then an object detection algorithm processes the large visual field photo (Figure 2.1) and produces the cropped image of a selected object. The detection and cropping tasks are performed on the server. The authors have not conducted an end-user study yet, nor have they provided detection accuracy scores. However, it is worth mentioning that, as in my study, the YOLO deep learning algorithm was

Figure 2.1: Image taken by omnidirectional camera
Note: https://pixabay.com/photos/360-degree-spherical-photo-office-1524199/, accessed 13/10/2022

used for object detection. A WiYG system presented by Feiz et al. (2019) utilises a smartphone and a custom 3D-printed attachment to guide blind users in filling out printed forms. The system tracks the user's signature guide and provides audio instructions to navigate to different form fields. The study reports an accuracy of 89.5% which suggests that the system effectively guided users to the correct locations on the form.

To sum up, state-of-the-art camera aiming assistive tools are based on AI solutions. Since object detection is a computationally intensive process and it is quite challenging to implement it on a mobile device, cloud-based technology is utilised. With this approach, some other issues related to privacy and the requirement for continuous internet connection arise. After all, continued efforts are needed to solve this problem and make a camera aiming more accessible to the VIP.

2.2.4 Combination of Tactile and Audio Feedback

As stated previously, users of TG prefer some form of non-visual feedback to increase accessibility. This section presents the most widely used feedback modalities for annotating TG, including tactile feedback, audio feedback and their combinations.

The **tactile** or haptic response is a type of feedback which is sensed by direct touch or applied to the user in the form of forces, vibrations or motions. An example of this type of feedback in the context of TG is Braille text legends that are embossed on the surface of the graphics. Usually the Braille text annotations are placed near the TG elements they correspond to (Figure 2.2). Braille text is a classic method for the labelling of graphics elements and was used for this purpose since the foundation of TG. One of the limitations of this approach is a decreasing Braille literacy trend. According to the statistics, the percentage of VIP who can read Braille is

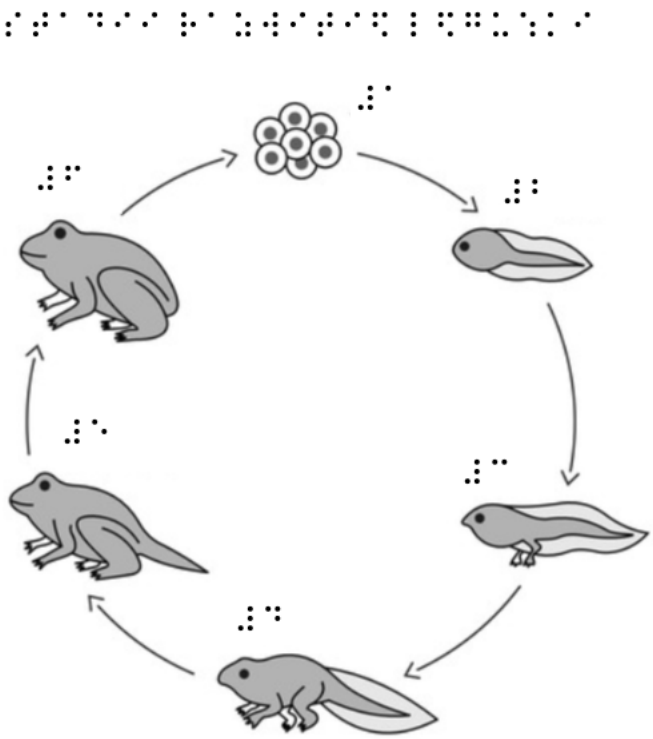


Figure 2.2: Tactile Graphics with Braille Text Annotations

12% (Institute, 2010). This number is around four times lower than it was in the 1960s when half of the legally blind population were Braille readers (Brittain, cited in Scheithauer and Tiger (2012)). However, it is not clear whether this number pertains to the United States or the global population, as the authors do not provide this information

Speech and non-speech **audio annotations** is a widely used method for labelling TG. This approach became more popular with the rise of Text-To-Speech (TTS) synthesizers. The TTS synthesizers allow computers or other machines to read text out loud in a real or synthetic voice. These synthesizers also resulted in computer and mobile screen readers advancements. According to statistics, the number of visually impaired computer users has risen from 65% in 2012 to 78% in 2016. Whereas, the number of mobile device users increased from 33% to 69% in the same years (Ofcom, 2017). These numbers match with the Brulé et al. (2020) findings. In their survey, the authors report that audio feedback is a primary interaction method for information acquisition among VIP. Compared to the Braille legends, the audio output does not require any additional skills to acquire the information and, therefore, is more accessible.

The use of a **combination** of audio and haptic feedback is a widely accepted method for designing accessible systems. According to research (Ross and Blasch, 2000), the implementation of tactile cues in conjunction with speech audio output has been found to be the most accessible interface for VIP. A key limitation of the study was the limited demographic scope of the participants, with ages ranging exclusively from 62 to 80 years. Multiple authors have used this method in their systems to enhance accessibility and usability (Gemperle, Ota, and Siewiorek,

2001; Olmschenk et al., 2015; Shilkrot et al., 2015; Cavazos Quero, Iranzo Bartolomé, and Cho, 2021).

In this section, various feedback tools for TG usage were discussed. The most widely used ones are Braille texts and pre-defined audio annotations. The combination of audio and haptic feedback is better than using either modality alone because it allows for a more interactive experience. While audio feedback can provide access to information, haptic feedback can provide additional cues about the structure and organisation of the information.

2.2.5 Barriers to the wider use of Tactile Graphics

There are certain barriers which hinder the wider utilisation of the TG in education. First of all, the process of creating such graphics is labour intensive. Sheppard and Aldrich (2001) report that according to the responses collected from the teachers (n=24), labour-intensiveness is the most cited issue associated with TG production. The teachers also claim that it is hard to make high quality TG to learners. Notably, special skills and experience are required not only to produce the TG but also to reduce the information overload and clutter within the image. For example, recent research suggests that the creation of interactive TG is challenging for teachers, social workers and VIP carers because it requires the knowledge of computer basics and vector-graphic software in particular (Thévin et al., 2019).

Another limitation related to the graphics is its production cost. Traditionally, the hardware used for TG production includes Braille embosser with Braille paper or swell paper with heat fuser. Dias et al. (2010) analysed the market and found that the lowest-cost Braille embosser is available for $690 (£575). Whereas, the price of the swell form machine (fuser) is $1391 (£1158). Either of the machines could be used depending on the production method. In addition, TG developers might have to pay for software that enables an easier production process. For instance, full access to the TactileView Design Software[2] costs $295 (£246). This product is also available for $60 (£50) on a monthly based subscription.

As previously reported in the literature, another concern related to the acceptance of TG is a mental load and a lack of special reading skills among students. For instance, Berla (1972) reported that the tactile shapes identification ability of blind students was low in elementary and middle school. In a further study, Berla and Butterfield Jr (1977) found out that special training in TG exploration strategies is required for students to achieve better ability and speed in detecting the objects on a tactile map. As a result, the lack of ability and experience in TG exploration requires teachers to instruct and preview the graphics with the student individually. This trend has been explored in the work by Zebehazy and Wilton (2014c), where 75% of TG users (n=59) find it useful when someone orients them to the tactile image. Consequently, more time is spent covering the material. Only 22% of the teachers agreed or strongly agreed that they had enough time to teach students how to use these graphics (Zebehazy and Wilton, 2014b).

[2]https://thinkable.nl/product/tactileview-software-licence/

To conclude, a demanding physical and intellectual effort, together with the high production price, are the main reasons which prevent TG from a wider usage. In addition, deficiency of TG reading skills and a significant drop in the Braille literacy among the learners, mentioned in the previous section, contributes to the issue.

2.2.6 Summary

In this section, the significance of the graphic literacy skills for VIP and the role of TG in their education were reviewed. One of the main limitations observed in the reviewed studies is the lack of background information provided for participants, including visually impaired students and their teachers. This missing context could potentially hinder the interpretation and generalisability of the findings.

Furthermore, various feedback methods designed to facilitate accessible exploration of TG were evaluated. Notably, the combination of audio and haptic feedback was found to be the most effective. This study also identified limitations in the existing solutions, including high production costs and the labour-intensive process of creating TG. A comprehensive review of TG conducted by Mukhiddinov and Soon-Young (2021) shows that despite significant advances in technology, the conventional methods for generating TG have persisted for many years. These findings underscore the need for continued research and development aimed at addressing these challenges and enhancing the accessibility of TG.

2.3 Existing Fingertip Detection and Tracking Methods

2.3.1 Overview

Fingertip detection is been applied in many fields. Especially, it plays an important role in facilitating more intuitive human interaction with a machine. Besides the Human-Computer Interaction (HCI), these solutions are used in the Virtual Reality (VR) object manipulation, hand gestures, sign language recognition, and the others.

In the classic CV, fingertip detection process generally consists of two steps. First, the algorithm performs hand segmentation. By this, the image area that is going to be processed is reduced; thus, a fingertip detection stage becomes less computationally expensive. After the hand region is successfully identified, fingertip detection is initiated. This task is very challenging due to the fingertip classes' similar appearance. This section provides a description of existing fingertip detection algorithms, along with their limitations. By exploring the strengths and weaknesses of these methods, this study aims to identify opportunities for improvement and the development of more effective fingertip detection technique. Table 2.1 presents existing hand segmentation and fingertip detection methods.

Reference Year	Hand region segmentation method	Fingertip detection method	Feature Vector	Back ground	Accuracy	Application	Key feature
Wu, Li, et al. (2017)	CNN	CNN	Color, shape, texture	Complex	99%	HCI	Color
Wu and Kang (2016)	YCbCr value of skin color	Calculating maximum distance	D between hand centroid and fingertip	Uniform	90.5%	HCI	Color
Baldauf et al. (2011)	RGB value of skin color	Calculating maximum distance	D between hand centroid and fingertip	Complex	-	Mobile AR applications	Color
Mukherjee et al. (2019)	Faster R-CNN, YCbCr of skin color	Maximum Curvature Points	Coordinates of MCPs	Complex	73.1%	Air-writing	Color and Features
Kounavis (2017)	Thresholding	Deformable Template Matching	Edge map & distance map	Complex	82.82%	Various applications	Template matching
Bhuyan, Neog, and Kar (2012)	Bayesian rule based skin color	Calculating minimum distance	Geometrical features of the fingers	Uniform	93.37%	HCI	Color based
Kim and Lee (2008)	RGB value of skin color	Template matching	Shape	Complex	90.5%	Navigation in 3D VR space	Template matching
Nguyen, Pham, and Jeon (2009)	RGB value of skin color	Thresholding	Shape	Complex	90-95%	HCI	Color and Feature
Qin, Zhu, et al. (2014)	Depth Thresholding	Convex hull	Maximum distance	Complex	91.9%	HCI	Shape
Fang et al. (2007)	HSV value of skin color	AdaBoost-based detector	Scale-space features	Complex	84-91%	HCI	Color and Feature
Gurav and Kadbe (2015)	HSV value of skin color	Haar Cascade and AdaBoost	Haar-like features	Complex	-	HCI	Color and Feature
Zhang, Liu, Zou, et al. (2018)	Manually annotated	HOG and LBP	Pixels gradients	Uniform	95-98%	HCI	Feature

Table 2.1: Hand segmentation and fingertip detection methods

2.3.2 Colour based

Detecting objects through colour or fiducial markers is often considered the simplest method for object detection. By knowing the boundaries of the marker colour space (RGB, HSV, etc.) or pattern of the fiducial marker attached to the finger, any fingertip class can be easily tracked. In the study by Bahrin, Yusof, and Na'im Sidek (2019) each fingertip class was tracked individually using five different colour markers. The gathered data was used to analyse how VIP explore tactile images. Zaman et al. (2016) used a glove with markers of three different colours for the detection of sign language letters. The same method with the gloves and markers was utilised by Mazumder, Nahar, and Atique (2018). The calculated angle ratios between the fingers were then used for the gesture recognition task. Chan, Yu, and Wong (2018) proposed a text region detection algorithm which was based on fiducial markers. The paper ring with an Aruco marker attached to it, worn on the index finger, helps the system identify the location of the fingertip. To sum up, the computational cost of this approach is very low, however, poor lighting conditions and the presence of an object with a similar colour strongly affect its accuracy (Sarkar, Sanyal, and Majumder, 2013).

2.3.3 Geometrical shape based

Geometrical shape based methods utilise the geometrical properties of the hand to detect the fingertips. Hand convexity, length, edges and centroid are the most common features which contribute to such detection. Ayala-Ramirez et al. (2011) used multiple geometrical features of the hand to recognise gestures in real-time. The coordinates of the fingertips were found by using geometrical features of the fingers (angle between fingers, distance between fingertips) in the work done by Bhuyan, Neog, and Kar (2012). In several works (Baldauf et al., 2011; Qin, Zhu, et al., 2014; Wu and Kang, 2016) fingertips were detected by identifying the maximum distances between the centroid of the hand and the edges. The primary limitation of this approach is its inability to effectively handle cluttered backgrounds (Hasan and Mishra, 2012).

2.3.4 Template matching based

Template matching is the process of finding areas in the image that are similar to the predefined template (patch). This is done by moving the template image over a bigger source image and calculating the differences between the pixels. This approach was used in a work by Yang, Jin, and Yin (2005). The authors used circular features of the fingertips as a template to detect the pointing gesture of the index finger. Same approach was used in the research conducted by Kim and Lee (2008) to find out the direction pointed by the user's finger. Similarly, Kounavis (2017) applies multistage template matching to detect fingertip contours in real-time. Template matching technique with the k-nearest neighbours (KNN) classifier was used to build a fingertip-writing character recognition system (Shih, Lee, and Ku, 2016). Similar to the previous approach, a

significant limitation of this method is its reduced efficacy in accurately detecting objects in cluttered backgrounds (Hasan and Mishra, 2012).

2.3.5 Motion based

Generally, motion-based analysis follows the object detection step. Granted that the object has been successfully detected in the frame, the algorithm uses this information to predict its location on the next one. Since the object detection phase is much more computationally expensive, this combined approach is more advantageous in mobile and embedded systems where processing capacities are low. Motion-based fingertip detection has been utilised in a few works. Oka, Sato, and Koike (2002) calculated the positions and the velocities of each fingertip and then applied a Kalman filter to foresee their locations on the next frame. Similarly, the positions and velocities of the fingertips together with the accelerations were taken into account to predict their locations in a consequent frame by Wang and Yuan (2014). Wu and Kang (2016) estimated fingertip motions by identifying finger curvature points and then using bidirectional optical flow algorithm. Prior to this step, fingertip detection was performed through geometrical analysis method, as described in earlier paragraphs. The primary constraint associated of this approach is its inability to detect the object once the tracking algorithm loses track of it. Therefore, an alternative method must be employed to re-detect the object.

2.3.6 3D model based

A kinematic 3D model of the hand is a mathematical representation of the hand's skeletal structure, joints, and range of motion. By utilising this model, the position and orientation of the hand can be accurately estimated in real-time based on the movement of its constituent parts, including fingers. This is achieved by comparing characteristic points obtained from depth or stereo cameras to the 3D model and determining whether they can be fit within the model's degrees of freedom. Once a digital model of the real hand is constructed, it is continually compared to the actual hand for matches, enabling accurate tracking of the hand's position and motion over time. This allows a more precise hand and fingertip detection. Liang et al. (2013) used a 3D model-based approach for hand pose estimation. After the 3D locations of the fingertips are detected, an inverse kinematics solver is applied to reconstruct the hand model. Son et al. (2016) proposed a fingertip detection method for the human-projector interaction. First, the hand region is identified and then all extreme points are extracted. Finally, these potential fingertip points are matched to the 3D complementary fingertip model and algorithm selects best candidates. This approach was also utilised in the following works (La Gorce, Fleet, and Paragios, 2011; Lu et al., 2020; Heap and Hogg, 1996). Dependence on specialised hardware is the main limitation of this approach. Also, this technique can only be used to detect objects for which a 3D model is available.

2.3.7 Feature classifier based

Feature classifier-based object recognition consists of two steps. First, various features are being extracted from the input image. Machine learning techniques such as AdaBoost, Haar Cascade, histogram of oriented gradients (HOG) and others can be utilised in this step. After that, the extracted features are fed into the classifier component. There an algorithm can determine which object classes have been detected and return probability scores along with their locations. These results can then be filtered based on their level of confidence to reduce the number of false positive detections. The region proposal component can be added before feature extraction to enhance the overall performance of the system. AdaBoost is a powerful real-time object recognition method. This algorithm turns weak classification learners into strong ones by constantly updating the weighted sum of each classifier according to its performance accuracy. Fang et al. (2007) utilised the AdaBoost algorithm to trigger hand gestures tracking and recognition in real-time.

Sometimes researchers implement a combination of algorithms, as was done by Gurav and Kadbe (2015). In their work, Haar-like features were extracted from the input image and an adaptive boosting algorithm was used to improve the accuracy of the system by choosing strong classifiers from each cascade of stages. Another example is proposed by Zhang, Liu, Zou, et al. (2018), where fusion of HOG and local binary patterns (LBP) algorithms was used to recognise hand gestures. A significant drawback of feature classifiers is their reduced performance in scenarios where lighting conditions vary (Sarkar, Sanyal, and Majumder, 2013).

2.3.8 Deep learning based

A Deep Learning (DL) model in CV is a neural network architecture designed to learn and extract meaningful features from visual data, which can be used for tasks such as object detection or image classification. There are many different DL based algorithms and approaches that are used for fingertip detection. For instance, the hand tracking component of Google MediaPipe (Zhang, Bazarevsky, et al., 2020) is a DL model that can be used to detect and track the location of the hands (including fingertips) in an input video stream. It uses a DL model trained on a large dataset of labelled images and videos. This model is able to learn the visual appearance of hands and their movement in space, allowing it to accurately detect and track hands in new unseen input data. The hand tracking component can be used in a variety of applications, such as virtual and augmented reality, gaming, and human-computer interaction. Figure 2.3 illustrates how the hand is tracked using this tool. However, solutions which use this model for fingertip detection were only able to run on PCs (Bahrin, Yusof, and Na'im Sidek, 2022), and those designed for mobile devices were unable to operate in real-time (Miwa et al., 2020).

Alam, Islam, and Rahman (2022) employed a Convolutional Neural Network (CNN) method to identify gestures and detect fingertips in their work. Specifically, the authors utilised the

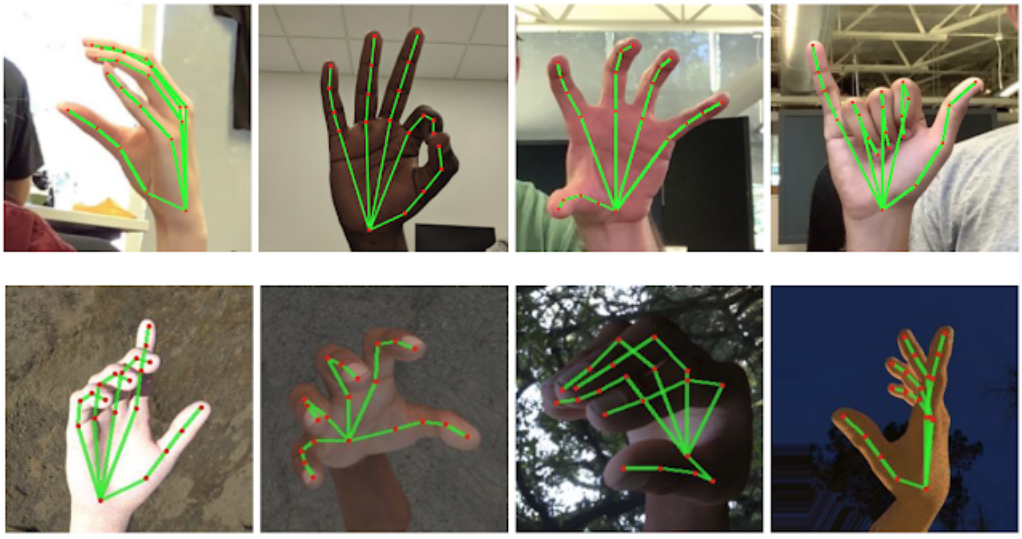


Figure 2.3: Hand tracking using Mediapipe
Note: https://google.github.io/mediapipe/solutions/hands.html, accessed 02/02/2023

YOLO9000 algorithm (Redmon and Farhadi, 2017) to train their model, using the EgoGesture dataset (Wu, Li, et al., 2017) as a source of training data. The results indicate a high level of accuracy for the developed model, and it is capable of real-time performance on a PC. However, it would have been more relevant if the authors had tested the model on a mobile device.

The Airpen model (Jain and Hebbalaguppe, 2019) is specifically designed for use with smartphones and hand-mounted devices. To train the model, the authors utilised the MobileNetV2 architecture (Sandler et al., 2018) and the mentioned EgoGesture dataset. While the authors assert that the model can operate in real-time, it is important to note that its current speed of 9 frames per second falls short of the typical minimum requirement of 15 fps for real-time performance (Angelova et al., 2015). Furthermore, the model's main drawback is its limitation to single-finger detection, as its performance degrades when multiple fingers are within its field of view.

2.3.9 Summary

Fingertip detection is the process of identifying the locations of the fingertips in an image or video of a hand. This can be useful for a variety of applications, such as virtual and augmented reality, gaming, and human-computer interaction. There are many different methods and algorithms that can be used for fingertip detection, including classic CV techniques and DL approaches.

Classic CV techniques involve manually extracting features from the input images and using these features to determine the locations of the fingertips. These techniques can be effective, but they often require careful design and tuning of feature extraction and detection algorithms. Additionally, these techniques can be sensitive to variations in the quality or appearance of input

Detection Method	**Advantages**	**Disadvantages**
Colour based	Computationally inexpensive and easy to implement	Low accuracy in poor lighting conditions and in the presence of an object with a similar colour
Geometrical shape based	Computationally inexpensive and easy to implement	Low accuracy in cluttered backgrounds and occlusions
Template matching based	Easy to implement, works well for simple patterns	Sensitive to variations in lighting and cluttered background, computationally expensive
Motion based	Effective for moving objects	Limited to moving objects, requires consistent background
3D model based	High accuracy in different lighting conditions	Requires special hardware and 3D model, computationally expensive
Feature classifier based	Fast detection speed	Sensitive to variations in lighting and cluttered background, requires labelled training data
Deep learning based	Robust to changes in appearance, works well for complex scenes, can detect multiple objects	Computationally very expensive, requires a large amount of labelled data

Table 2.2: Pros and cons of various detection methods

images, making them less robust than DL approaches.

DL techniques involve training large, complex neural networks on a dataset of labelled images of hands. These networks can learn to automatically extract features from input images and use these features to identify the locations of the fingertips. DL techniques can achieve better performance and more robust results than classic CV, especially when working with complex, real-world data. Additionally, DL techniques can be more efficient and scalable, making them well-suited for large-scale applications such as fingertip detection in video streams. Table 2.2 presents advantages and disadvantages of the methods mentioned above.

2.4 Educational Systems for Visually Impaired

2.4.1 Overview

In this section different types of educational systems for VIP are presented. The systems comprise solutions that facilitate the acquisition of information mainly through the sense of touch. On the basis of the hardware requirements, these systems were divided into four sections. First, touch screen based tools are evaluated. Then systems which require computers with a web camera are described. After that, depth camera based solutions are assessed. Finally, mobile phone and tablet based approaches are discussed.

2.4.2 Touch screen based

Nomad (Parks, 1988) audio-tactile tool was developed in 1988 and the system was the first-of-its-kind. The device was connected to the computer, and a touch-sensitive surface was used to trigger corresponding predefined audio descriptions. The system was commercialised the following year but was not very popular among users. The poor acceptance of the device was not directly associated with its design but was rather determined by the overall technological state of that time. The screen resolution and user-friendliness of the speech synthesizers back then were not as high as they are nowadays. The Nomad system laid the foundation of the audio-tactile based technology era and some of its most successful descendants are described in the next paragraphs.

TTT (Talking Tactile Tablet) (Landau and Gourgey, 2001) is one of the first audio-tactile systems which has been used in educational institutions. First, the user has to place the printed TG on the tablet display. After that finger press events are sensed and a signal is being sent to the computer. The special program uses this information to trigger predefined audio information about TG based on the touch coordinates. This device was developed in the 2000s and was successfully commercialised. The current market price of the TTT is \$799[3] (£665).

ViewPlus IVEO (Gardner and Bulatov, 2006) is another proprietary system that initially started as a research project at Oregon State University and then was commercialised by ViewPlus. The overall hardware design is similar to TTT and the user interacts with the TG through the high-resolution touchpad. Complementary software editor for Scalable Vector Graphics (SVG) creation is provided as well. Both components are available online and cost \$1129 (£939). High cost and portability are the main limitations of these proprietary devices.

Refreshable Tactile Graphics Applied to Schoolbook Illustrations (Petit et al., 2008). Researchers from the University of Montreal developed a system based on a refreshable tactile graphics device called STReSS2 (Stimulator of Tactile Receptors by Skin Stretch). This device produces tactile feedback by creating small surface vibrations. Furthermore, the authors utilised

[3]http://touchgraphics.com/portfolio/ttt/

the MaskGen application to transpose the images from the school books to the STReSS2 tactile display. They have conducted experiments with twenty visually impaired and twenty sighted people with blindfolds to test the system. The participants explored three types of tactile images and answered questions related to their contents. Overall, the scores for the correct answers were between 70 and 80 percent. Limitations of the system are: cost of the refreshable device and the small size of the exploration area.

Name	**Hardware**	**Software**	**Tactile Output**	**Interaction Modes**	**Limitations**
Nomad (Parks, 1988)	Electronic touchpad connected to a computer	Nomad Kernel, CAD, Information Access System, Walkabout System	Swell paper	Direct touch	High cost, not portable
TTT (Talking Tactile Tablet) (Landau and Gourgey, 2001)	Electronic tablet connected to a computer	World Map, TTT Match Game, TTT Snakes & Ladders, TTT Tool	Swell paper	Direct touch	High cost, not portable
IVEO by ViewPlus (Gardner and Bulatov, 2006)	Electronic touchpad	IVEO Player Pro	Braille paper	Direct touch	High cost
Refreshable Tactile Graphics (Petit et al., 2008)	STRESS Refreshable tactile display (Pasquero and Hayward, 2003)	Xenomai, MaskGen, Photoshop	Refreshable tactile display	Direct touch	High cost, exploration area small size

Name	Hardware	Software	Tactile Output	Interaction Modes	Limitations
IMG (Interactive Multimodal Guide) (Cavazos Quero, Iranzo Bartolomé, and Cho, 2021)	Enclosure display, Arduino, proximity touch sensor	ZW3D 3D drawing software	3D printed model	Direct touch	High cost

Table 2.3: Touch screen based systems

2.4.3 Computer and web camera based

Tactile Graphics Helper (TGH) is a system that utilises CV algorithms to track a user's fingertips (Fusco and Morash, 2015). The prototype runs on a computer and uses a mounted camera to acquire images. In addition, researchers developed Matlab-based Graphical User Interface (GUI), which allows to create a tactile image accompanying a file. The list of all objects and their descriptions are stored in this file. After the data is uploaded to the TGH system the user can start exploring tactile image and listen to TTS generated audio information. In addition to finger-pointing feature, voice commands are available as well. The limitations of the project are its portability issues (computer and camera) and the CV dependence on a surroundings light condition.

Access Lens (Kane, Frey, and Wobbrock, 2013) enables users to read texts from physical objects by pointing the index finger upon them. CV software detects the fingertip and recognises the text via a web camera. The system has three different interaction modes: direct touch, virtual edge menus and voice commands. The direct touch mode detects the text which is closest to the fingertip and speaks it out loud. When edge menus are activated AL adds virtual buttons with lists of recognised texts in alphabetical order so it is more convenient for the user to access them. In the third mode, the AL is controllable via various voice commands. For instance, command "List" lets the AL speak out the list of all detected texts. Overall, users were satisfied with the system and its different modes. The disadvantages of the system are its portability constraints and strong CV and OCR (Optical Character Recognition) dependence on the lighting conditions.

Shamsul Bahrin, Md Yusof, and Na'im Sidek (2022) present a laptop-based **TG reading device** that uses the Google MediaPipe algorithm (Zhang, Bazarevsky, et al., 2020) to track the user's hands and provides audio feedback to VIP. Although the proposed system offers significant advantages, such as natural and real-time interaction with TG, its main limitation is the lack

of portability. Additionally, the system has not yet been tested with end-users, highlighting the need for further research to evaluate its effectiveness.

Name	Hardware	Software	Tactile Output	Interaction Modes	Limitations
The Tactile Graphics Helper (TGH) (Fusco and Morash, 2015)	Laptop and webcam	Matlab	Swell paper	Finger pointing and voice commands	Not portable, sensitive to variations in lighting
Access Lens (AL) (Kane, Frey, and Wobbrock, 2013)	Laptop and webcam	OpenCV, Microsoft NET speech library, ABBY Fine Reader	Regular paper	Finger pointing, virtual edge menus and voice commands	Not portable, sensitive to variations in lighting
Tactile Graphics Reading Assistive Device (Shamsul Bahrin, Md Yusof, and Na'im Sidek, 2022)	Laptop and webcam	MediaPipe (Zhang, Bazarevsky, et al., 2020)	Swell paper	Finger pointing	Not portable

Table 2.4: Computer and web camera based systems

2.4.4 Depth camera based

Markit and Talkit (Shi, Zhao, and Azenkot, 2017) is a system which allows the user to create audio annotations and then interact with 3D-printed models. Markit is a user interface that is developed for instructors and teachers of VIP and is used for creating, designing and annotating 3D models. Talkit is an associated toolkit that is used for the information interpretation. RGB camera senses the spatial locations of the sticker that is attached to the user finger and the special marker. The marker, mounted on the model, acts as a reference for the CV algorithm and helps it to correctly identify the relative positions of the sticker and the 3D object's labelled regions. The experimental results showed that interaction with 3D models is very intuitive for VIP and

on average it takes less than 8 seconds for them to find a specific annotation. The limited viewing angle of the camera and its performance under poor lighting conditions are the main disadvantages of the system.

CamIO (Camera Input-Output) (Shen et al., 2013) is a system which utilises a Microsoft Kinect camera to track the spatial positions of a user's fingers, enabling the detection of their interaction with physical objects. The objects must be placed on a flat surface with fiducial markers, which serve as a reference for the CV algorithm. The system triggers TTS-synthesised audio feedback upon touch events. While the system has demonstrated the potential for effective interaction between users and physical objects, it also has some limitations. The primary drawbacks include a lack of portability and high cost associated with the required hardware (Kinect camera and laptop).

IAG (Interactive Audio Guide) (Reichinger et al., 2016) employs an RGB-D camera, specifically the Intel RealSense F200, to detect users hand gestures. The depth value recorded by the camera allows for tracking of both on- and off-object interactions, enabling the system to recognise multiple combinations of gestures and trigger various audio feedback accordingly. Researchers evaluated the system in a 'museum-like' setting where potential users interacted with the tactile reliefs of paintings. While the system shows promising results, limitations include the high cost of the depth camera and the system's lack of portability.

Name	**Hardware**	**Software**	**Tactile Output**	**Interaction Modes**	**Limitations**
CamIO (Camera Input-Output) (Shen et al., 2013)	Microsoft Kinect Camera	Computer vision algorithms	3D-printed model	Direct touch	Not portable, high cost
IAG (Interactive Audio Guide) (Reichinger et al., 2016)	Intel RealSense F200	Computer vision algorithms	3D relief surface	Finger pointing	Not portable, high cost

Name	Hardware	Software	Tactile Output	Interaction Modes	Limitations
Markit and Talkit (Shi, Zhao, and Azenkot, 2017)	RGBD camera	OpenCV	3D-printed model	Direct touch	Not portable, limited viewing angle of the camera, sensitive to variations in lighting

Table 2.5: Depth camera based systems

2.4.5 Mobile phone and tablet camera based

TPad (Melfi et al., 2020) is a mobile educational application based on the tablet (IPad Pro) that allows users to explore tactile images. Users must place a printed tactile image over a tablet screen and start exploring it with their hands. The system will provide audio clarifying feedback by detecting touch instances. Also, a 3D-printed plastic frame is required to hold the A4-sized paper still. The TPad uses SVG files with information about the image and objects it includes to provide audio descriptions. Different modes allow users to download preprocessed SVG files easily. For example, by scanning QR codes located on the backside of the image. In addition, researchers developed a special web interface for instructors. There, they can upload and organise the graphics and send them to the server to where all the TPads are connected. During the experiments, the system showed that its users acquire information about TG in a faster manner with a 70% accuracy. The drawbacks of the system are the cost of the tablets and potential difficulties with the frame 3D printing since some educational organisations do not have them.

Tactile Graphics with a Voice (Baker et al., 2014) is an application that runs on the smartphone and provides feedback by scanning QR codes placed on the TG. The main motivation of the study was to replace large Braille texts with more compact codes; thus, providing more information about TG. There are three different modes available to help the user properly aim a phone camera: silent, audio instructions and finger pointing. Instructions navigate the user to a QR code by sensing the phone orientation whereas finger pointing mode helps an app to identify the correct QR code when multiple labels are visible. Overall, users were satisfied with the system, but aiming a smartphone camera properly remains a challenging task for VIPs.

THATS (Touch and Hear Assistive Teaching System) is a mobile app that allows the user

to explore predefined tactile images by providing accompanying audio descriptions. In addition to the app, the THATS team developed an online editor and linked it to the digital library. In this way, instructors can either create TG from scratch or download ready-to-use ones. THATS implements widely used CV algorithms (background subtraction, image thresholding, etc) to detect the fingertip. To sum up, THATS is a very promising project which is designed to give visually impaired community free access to easy-to-use educational materials but the app has not launched yet and there are no experimental results publicly available.

Researchers from Cornell University created a **Molder** (Shi, Zhao, Gonzalez Penuela, et al., 2020) - an accessible tool for tactile maps design and exploration. This tool has four main components: a physical frame, a website for the model's creation, a mobile app for the model exploration and a server. The Molder was tested by the end users and results showed that participants with different vision abilities were able to create tactile models using this tool. The main disadvantage of the Molder is that it only supports a single-finger exploration. In addition, production of the 3D models is time-consuming and has a high cost.

TARS (Hosokawa, Miwa, and Hashimoto, 2020) is another mobile application which provides audio descriptions whilst the user explores a tactile image. This app utilises Google's MediaPipe (Zhang, Bazarevsky, et al., 2020) hand-tracking system for fingertip detection. According to their experimental results, the app detects fingertips with 85.5% accuracy. However, it was not clearly stated whether this result is for all five fingertips or just the index one. In addition, a single-frame processing time was not specified. In the further work conducted by the same authors (Miwa et al., 2020), it took two seconds to process a single frame. Whereas real-time execution expects at least 15 fps (frames per second) (Angelova et al., 2015). Another limitation of the study is that the authors have not described how the tactile images and corresponding annotations are created.

Name	**Hardware**	**Software**	**Tactile Output**	**Interaction Modes**	**Limitations**
Tactile Graphics with a Voice (Baker et al., 2014)	IPod touch and IPhone	QR code reader	Braille paper	Finger pointing guidance and voice guidance	Aiming the phone camera
TPad system (Melfi et al., 2020)	Tablet	non-CV	Regular paper	Direct touch	High cost of 3D printing and Braille embosser

Name	Hardware	Software	Tactile Output	Interaction Modes	Limitations
Touch and Hear Assistive Teaching System (THATS)	IPhone	Computer vision algorithm	Swell paper	Finger pointing	Aiming the phone camera
Molder (Shi, Zhao, Gonzalez Penuela, et al., 2020)	IPad, 3D printer	OpenCV, SpeechRecognizer	3D printed object	Finger pointing and voice commands	Single finger exploration, High cost of 3D printing
TARS (Hosokawa, Miwa, and Hashimoto, 2020)	IPhone	MediaPipe (Zhang, Bazarevsky, et al., 2020), AVSpeechSynthesizer	Swell paper	Finger pointing	Slow fingertip detection speed

Table 2.6: Mobile phone and tablet camera based systems

2.4.6 Summary

I have described various Assistive Technology (AT) solutions designed to make TG more accessible. Five of these systems run on a mobile devices. TPad, Molder and the Tactile Graphics with a Voice devices have already been tested and shown promising results. However, there are still some drawbacks, i.e. the high price of 3D printers or failure to detect multiple fingers. There are indications that work on the THATS device may have been discontinued and TARS is not able to process in real-time on a mobile device yet.

In light of the conducted review, I have identified four key criteria that characterise the design of the developed system. These criteria encompass:

- High accuracy under low light conditions
- Implementation on a mobile device
- Real-time execution
- Allow two handed exploration

It is essential to prioritise these criteria in the development process to ensure that the final product is optimised to meet the desired performance standards. It is worth noting that implementing the system on a mobile device offers several benefits, including portability and cost-effectiveness.

2.5 Conclusion

This chapter begins with a review of the literature regarding the importance of TG in VIP education. Also, various feedback modes for the TG exploration and their acceptance by the end-users are presented. This includes audio feedback, tactile feedback, and the combination of both. The section then explores the limitations and barriers which inhibit the wide usage of TG in education. According to the literature, labour-intensiveness and cost of TG production together with mental load on the reader are the main barriers which hinder a wider TG utilisation.

In addition, this section presents a review of recent literature on existing fingertip detection and tracking methods. A closer look at the literature on classic CV algorithms for fingertip detection revealed a number of limitations. For instance, methods which rely on colour markers or geometrical shapes perform poorly when lighting conditions change or in cluttered environments. On the other hand, existing DL based solutions lack the ability to perform real-time detection on mobile devices.

Finally, this chapter provides a review of the literature on educational systems for VIP. The advancements in technology have led to the increased computational capabilities of edge devices, shifting the focus of research from large, PC-based systems to small, mobile devices with integrated computing and cameras. After examining existing assistive systems that enhance accessibility to TG and allow for independent information acquisition without sighted assistance, their limitations were identified. This formed the basis for proposing a novel system that addresses these limitations.

Chapter 3

Fingertip Detection

3.1 Introduction

Fingertip detection is a crucial part of my research because it enables users to interact with Tactile Graphics (TG) in real time. This process involves identifying the position of the fingertips in an image or video and is essential for the effective use of the TAURIS system. The app I have developed tracks the movements of the user's fingers as they explore the image and provides information about what they are touching. Accurate fingertip detection is therefore a key component of the app's functionality. The technical background for the development of this Computer Vision (CV) algorithm can be found in Appendix A. In Section 3.2, I describe the development process of a model that was used in my app for detection. Finally, the whole chapter[1] is summarised in a conclusion section.

3.2 TAURIS Fingertip Detection Algorithm

In this section, a novel fingertip detection method that is specifically tailored to meet the needs of my research is described. The review of the existing methods was presented in Section 2.3 of the thesis. The developed algorithm facilitates more accurate and efficient detection of fingertips. I will provide a detailed description of the method, including its key components and the challenges it addresses. The experimental results demonstrating the effectiveness of the approach are provided as well. Overall, this novel fingertip detection method represents a significant advancement in the field with potential applications in a wide range of tasks.

After reviewing the literature, it was found that state-of-the-art Deep Learning (DL) models were capable of running on mobile devices and detecting objects accurately even in real-time (Howard et al., 2017; Sandler et al., 2018; Qin, Li, et al., 2019). Thus, it was decided to create a

[1]Some of the work in this chapter has appeared in Zeinullin and Hersh (2022). Maralbek Zeinullin is the first author and main contributor to this paper.

DL model tailored to meet our needs. There are three main components required to build a DL object detection model: training dataset, pre-trained model, and appropriate hardware.

Training dataset. Labelled images are the building blocks of any object detection model. Usually, the more images are fed into the model during the training process, the more capable model is produced. Most of the time, researchers utilise ready-to-use datasets from the internet. As a matter of fact, there are plenty of high-quality labelled images available online. For example, the ImageNet project is considered the largest visual dataset (Deng et al., 2009). This project is a crowd-sourced database containing more than 14 million labelled images for 20,000 different object classes. Other well known datasets used for the pre-trained model creation are: Open Images (Krasin et al., 2017), Microsoft Common Objects in Context (COCO) (Lin, Maire, et al., 2014) and PASCAL VOC (Everingham et al., 2010).

A pre-trained (or parent) model is a Deep Neural Network (DNN) trained on a large dataset for relatively long period of time. For example, it takes 14 days to train a ResNet-50 object detection model on a NVIDIA M40 Graphical Processing Unit (GPU) using the mentioned ImageNet dataset for 90 epochs (You et al., 2018). (Usually, it takes less than a day to train a customised model on top of the parent model). These models have the ability to generalise well on the images outside of the training set and are used as a starting point for customised model training. Pre-trained models are capable of detecting common object classes such as people, cars, animals, etc. If a customised model is required, researchers produce the datasets themselves and train a new model on top of the pre-trained model. The logic behind this approach is that the parent model is already capable of detecting basic features like angles, edges, shapes, etc. Therefore, there is no need to train the model to detect them again and instead enable the researcher to focus on training the model on more unique features. As a result, less time and resources are required to train a new model. This process is called transfer learning and is considered preferable to training a neural network from scratch.

Hardware. Depending on the task and the type of neural network architecture, the hardware requirements may differ. The training of a machine learning model can be done on a Central Processing Unit (CPU), but DL models with multiple layers require GPU. Shi, Wang, et al. (2016) in their work show that CPU training is about ten times slower than a GPU one. GPUs are designed to execute multiple calculations simultaneously, which makes them well-suited to the matrix calculations used in DL. As a result of increased demand, graphic cards have been in short supply in recent years[2]. Fortunately, there are services that provide free cloud servers. Google Colab, Kaggle, Amazon, and others give access to their GPU and Tensor Processing Unit (TPU) resources at zero cost. On the other hand, cloud computing involves storing and processing data on remote servers, which can make it more vulnerable to data breaches. There always will be privacy and security issues associated with it (Sun, 2020).

[2]https://www.forbes.com/sites/forrester/2021/05/06/the-global-chip-shortage-wont-ease-soon/

3.2.1 Detection Models

Overview

Object detection models are a type of DL models that are used to identify and localise objects within an image or video. In this section different Convolutional Neural Network (CNN) object detection model architectures are compared. Our main criteria for selecting models was their ability to detect objects in real-time while running on a mobile device. A substantial amount of research has recently been devoted to the development of such network architectures. Mainly, this trend was sparked by the exponential growth of self-driving cars, augmented reality applications and CCTV cameras. In an original YOLOv3 paper, SSD, RetinaNet and YOLOv3 family architectures showed the fastest detection rate with a tolerable accuracy trade-off (Redmon and Farhadi, 2018). Thus, these three models were selected for initial training and testing in my research. First, an overview of each model and its neural network architecture is presented. Then, their performances are compared and the model which best suits the research requirements is selected. Across all models, the same image set was used for training and evaluation. The training parameters for each model are presented in Table 3.1. Furthermore, inference times for all models were compared using the same hardware (NVIDIA Quadro RTX 5000 GPU).

Model	# of convolutional layers	Pooling Layer	Activation Function	Optimization Method	Learning Rate	# of epochs	# of training hours
MobileNet v2	28	Average Pooling	ReLU	Momentum	0.08	100	20
RetinaNet	50	Average Pooling	ReLU	Momentum	0.04	100	14
Tiny-YOLOv3	13	Max-pooling	Leaky ReLU	Momentum	0.01	100	8

Table 3.1: Models training parameters

SSD MobileNet V2

MobileNet (Sandler et al., 2018) is a CNN model designed to perform well on mobile devices. Although it has a deep network structure, the model is considered fast and efficient. This is achieved by utilising depthwise separable convolutions. The main difference of this convolution method from the standard one is that it splits the computation into two steps: depthwise and pointwise. Each input channel is convolved by a single filter in depthwise convolution. Whilst

in pointwise one, the output of the depthwise convolution is combined linearly. This allows MobileNet to achieve good accuracy while using fewer parameters and requiring less computation than traditional convolutional neural networks. The model architecture is presented in Figure 3.1. An input image with dimensions of 224x224x3 is 224 pixels wide, 224 pixels tall, and has three colour channels (red, green, blue). Each pixel is represented by three 8-bit integers, resulting in a total of 224x224x3 = 150,528 bytes of data for the entire image.

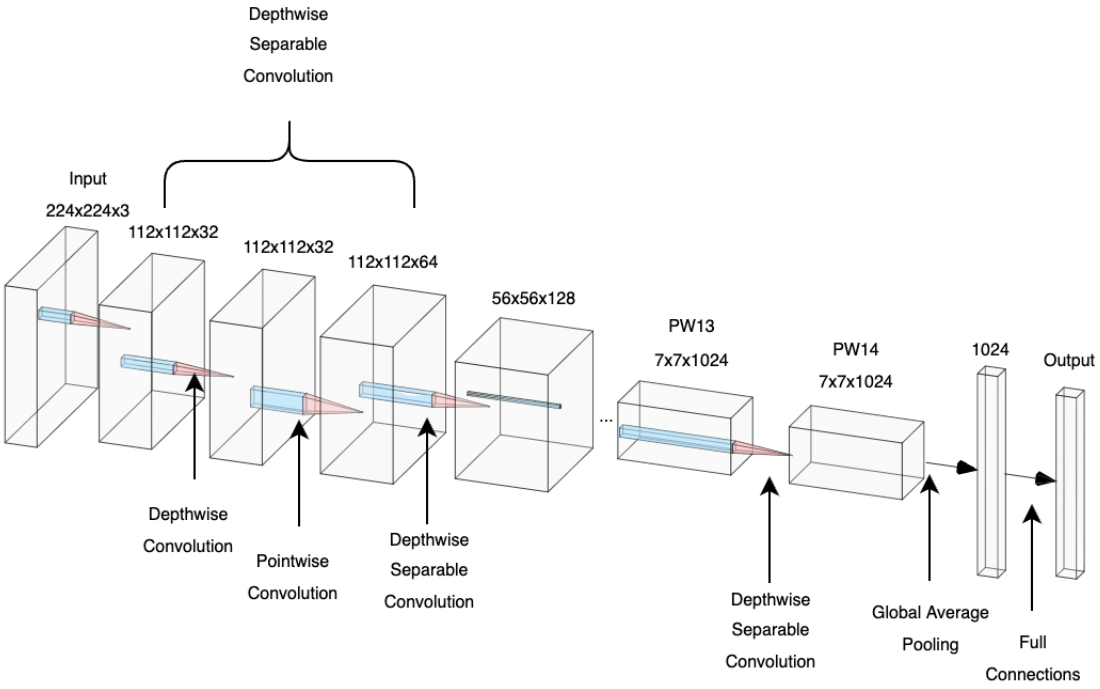


Figure 3.1: MobileNet V2 architecture
Note: Modified from source: Wang, Hu, et al. (2020), p.3

RetinaNet

RetinaNet is an SSD model that uses focal loss to address class imbalance during the training process (Lin, Goyal, et al., 2017). In other words, this model focuses on hard examples and ignores easy negatives, thus not hindering the detector during training. For instance, if you need a model to detect people, dogs and cats, the model will focus on the features which differentiate dogs and cats. These two look alike and their appearance is very distinct from that of humans. By this, fast and more accurate detection is achieved. The architecture of this model is based on a unified network consisting of a Feature Pyramid Network (FPN) (Lin, Dollár, et al., 2017) backbone and two task-specific subnetworks (Figure 3.2). The FPN backbone is used to compute feature maps and build a multi-scale feature pyramid. Afterwards, the first subnetwork performs classification and outputs the probabilities for each object class presented in the image. Lastly, the second subnetwork applies convolutional bounding-box regression to calculate the offset from ground-truth object boxes.

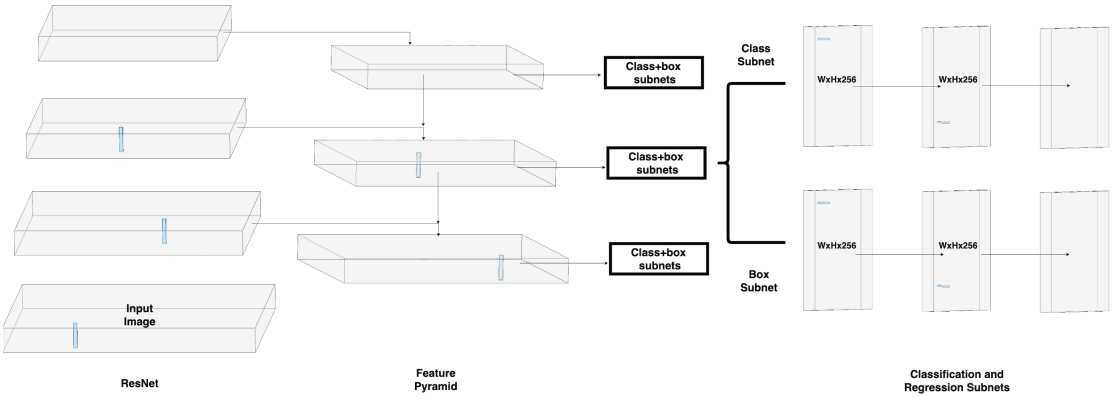


Figure 3.2: RetinaNet architecture
Note: Modified from source: Lin, Goyal, et al. (2017), p.5

YOLOv3

YOLO is a family of Computer Vision (CV) algorithms first introduced in 2016 (Redmon, Divvala, et al., 2016). YOLOv3 (Redmon and Farhadi, 2018) is the descendant of the YOLO algorithm. The main advantage of this model architecture is that it can process images in real-time. This is achieved by a unified architecture of the network. Tiny-YOLOv3 is a compact and accelerated version of YOLOv3 which was designed for embedded and mobile systems. The smaller architecture size makes the tiny-YOLOv3 extremely fast, the number of convolutional layers in YOLOv3 and its tiny version is 24 and 13 respectively. The architecture of the model can be found in Figure 3.3. Higher detection speed comes with the price of lower accuracy. Redmon, Divvala, et al. (2016) show that the smaller version works 3.5 times as fast with just a 10% accuracy trade-off.

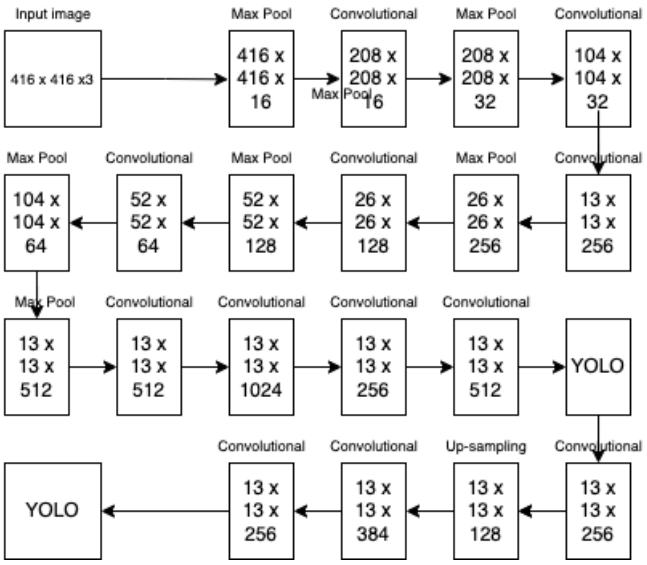


Figure 3.3: Tiny-YOLOv3 architecture

Comparison

After training and testing the mentioned models, Mean Average Precision (mAP) values and inference speeds were recorded. Table 3.2 summarises all the results. Average Precision (AP) is a metric that is used to calculate the performance of a detection model. We must first determine these four entities in order to calculate the model's accuracy value.

- **Confusion matrix**. A table summarising the performance of the object classifier. Figure 3.4 illustrates different instances used in this sub-metric.
- **Precision** . This sub-metric shows how accurate the model estimate was when it predicted the object. The formula 3.1 shows how it is calculated.
- **Recall**. It measures the proportion of positive instances that were correctly identified by the model. This sub-metric is determined using Formula 3.2.
- **Intersection over Union (IoU)**. It measures the overlap between ground truth and detected bounding boxes. See Figure 3.5.

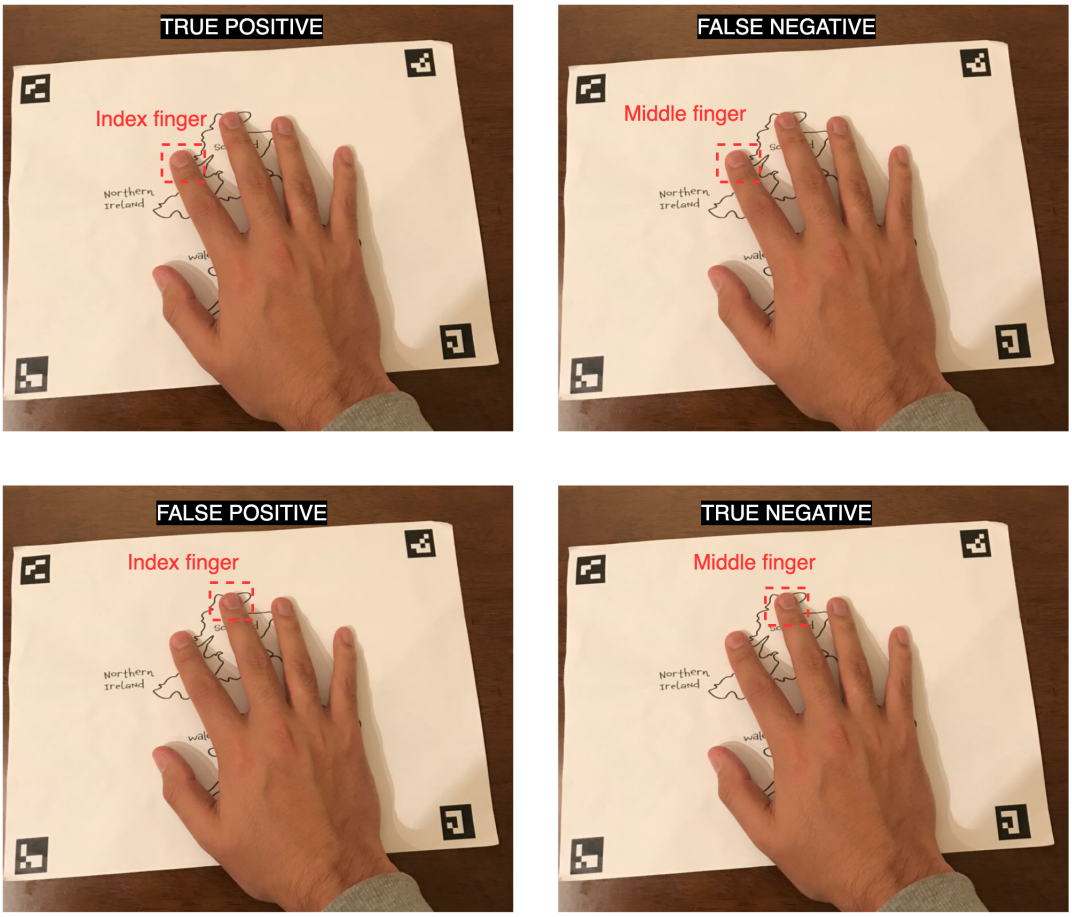


Figure 3.4: Confusion matrix

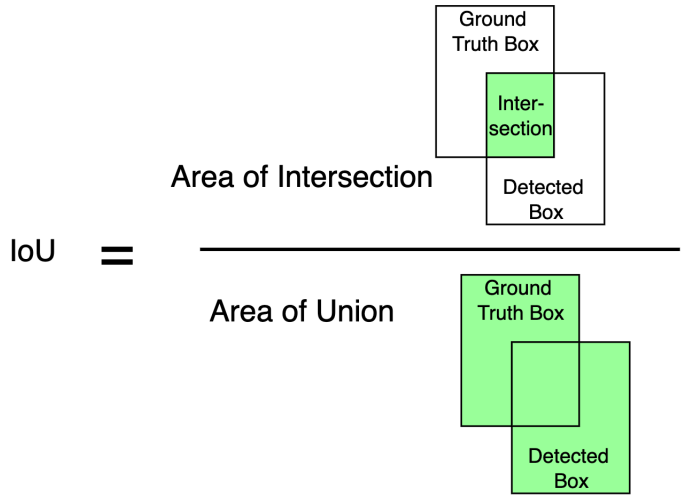


Figure 3.5: Intersection over Union

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{3.1}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{3.2}$$

In a few words, it takes into account the trade-off between precision and recall at different IoU threshold values and calculates the AP. Mean Average Precision, as its name suggests, is just the mean of AP values of the total number of classes (Equation 3.3).

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{3.3}$$

According to Table 3.2, all three models have almost similar mAP scores. On the other hand, their single-frame detection rates are very different. As was expected, tiny-YOLOv3 showed the fastest inference speed- more than 10x better than the second fastest model. It is worth mentioning that in the context of my research, the index finger detection is crucial. This is related to the fact, that most blind users prefer to use this finger to read Braille texts (Wong, Gnanakumaran, and Goldreich, 2011). From this, I assumed that the index finger will be leading the whole TG exploration process and it would be logical to track this fingertip. From the same table, it can be seen that tiny-YOLOv3 performance on this finger detection is 2.5 times lower than in the other two. However, considering the speed and accuracy trade-off, it was decided to continue the project using the tiny-YOLOv3 and build all future models on top of this lightweight model. So, it was important to boost the detection accuracy by improving the model with additional images and tweaking the overall detection algorithm. All these enhancements will be thoroughly discussed in the next sections. Again, the three models were trained using exactly the same images and tested on the same hardware.

Class	**MobileNet V2**	**RetinaNet**	**Tiny-YOLOv3**
Thumb	24.9	13.3	**39.9**
Index	50.11	**57.08**	22.04
Middle	44.37	**66.80**	39.61
Ring	60.7	59.88	**71.95**
Little	87.2	77.55	**89.63**
mAP@50 (%)	53.46	**54.92**	52.63
Inference speed (ms)	24.3	33.8	**2.13**

Table 3.2: Models accuracy and speed comparison

3.2.2 Datasets

The goal of our model was to identify the fingertips of both hands in images. We searched for datasets that could be used to train and evaluate the model and found the EgoGesture dataset (Wu, Li, et al., 2017). This dataset includes approximately 59,000 first-person view images. These images are divided into 16 different gesture categories, with approximately 3700 images in each category.

First, I experimented with the index finger pointing gesture. The drawback of this approach was that the algorithm often confused the index finger with other fingers due to their very similar appearances. Therefore, robust and accurate performance was achieved only when the index finger alone was visible on the frame. Since it is not very advantageous for the user to explore a tactile image with one finger (Leo, Cocchi, and Brayda, 2016), another model was trained using images of one hand only. This set was called SingleFive and can be found on Figure 3.7.a. The produced model was able to accurately detect the fingertips when only one hand was in the field of view. However, it was performing poorly when two hands were visible. Therefore, it was decided to train another model using the PairTen set. This set contained the images of both hands (Figure 3.7.b). In contrast to the previous model, it detected the fingertips correctly in cases where two palms are visible and failed when only one was present. Because I wanted the app to be flexible and convenient for the users, a model which can perform well in both situations was essential (when either one or two hands were present). Therefore, it was decided to merge the two image sets and train a third model. Third model successfully detected the fingertips during both scenarios and, thus, was used by the app during the experimental sessions.

TAURIS Dataset

The performance of the model trained on the EgoGesture dataset was effective during both tests (Table 3.3) and actual experimental sessions. However, it was clear that its performance could

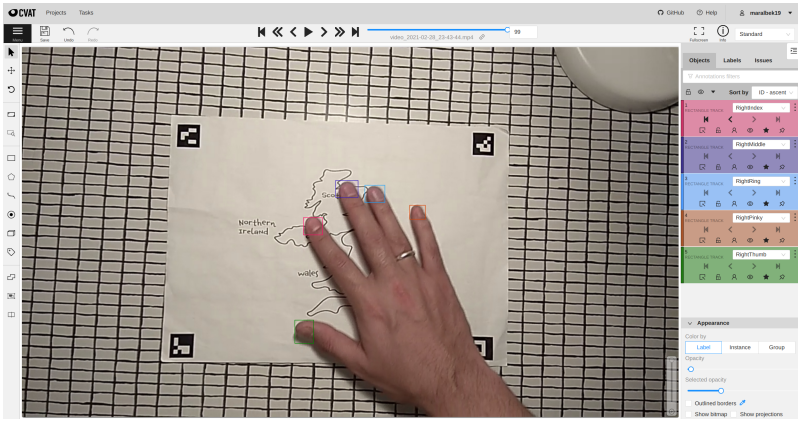
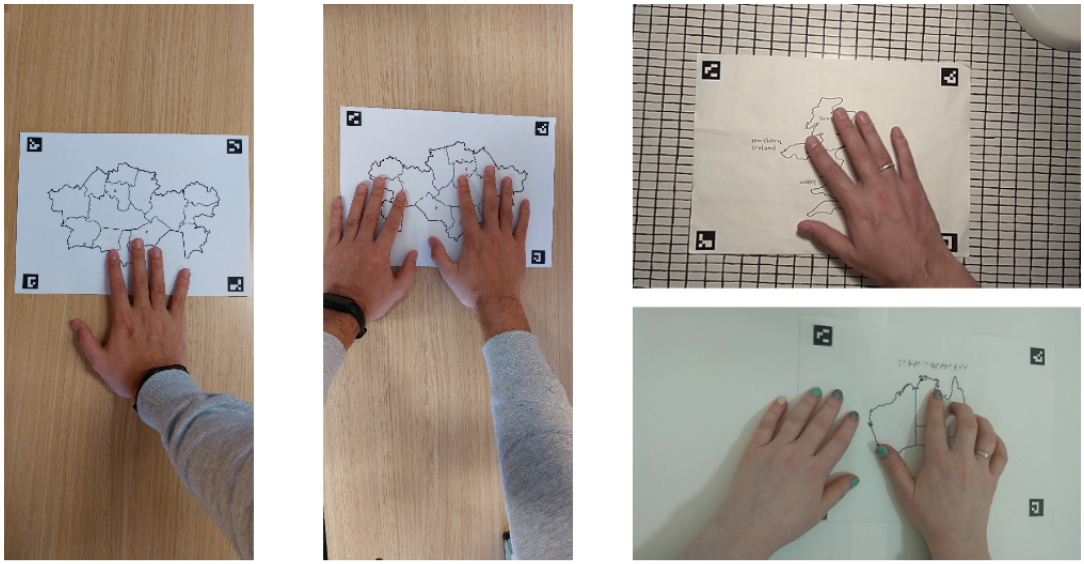


Figure 3.6: Image annotation

be improved with more images. For this purpose, the experimental sessions were video recorded for further labelling and model training. I informed all participants about this and they gave their consent. It is important to note that I recorded only their hands when they explored the TG. I did not capture images of faces or other parts of the body that could reveal their identity. From the recorded videos, 2000 frames were selected for annotation. While it is generally true that a larger number of images can improve the performance of a detection model, only 2000 images were selected due to the time-consuming nature of the annotation process. By annotation, I mean manually reviewing and labelling the images by drawing bounding boxes around each fingertip (as shown in Figure 3.6). The CVAT video annotation tool was used for this process (Sekachev et al., 2020). The training set contained 1000 images of SingleFive and another 1000 of PairTen gestures (Figure 3.7.c). By extending the existing dataset with new images, the model accuracy increased significantly (results presented in the next section). The whole process of datasets selection is illustrated in Figure 3.8.

Tiny-YOLOv3 Evaluation

Evaluation is an essential part of ML model development. This process is conducted using the test set. Usually, this set is produced by randomly splitting the whole dataset into two parts, 10% (test set) and 90% (training set) (Joseph, 2022). To obtain unbiased results, it is important that the images in the test set are different from those in the training one. In other words, the model should not see these images when it is being trained. In my research, it was crucial to create a test set which would act as a benchmark for the models trained during the whole research process. This test set is different from the TAURIS dataset and was created almost one year before. The use of this test set allowed the researcher to monitor the performance changes of the models produced throughout all stages. To create a test set that is close to the real-life setting, the researcher recorded himself while exploring the TG. In total, 200 images were collected and annotated. Results presented in Table 3.2 and Table 3.3 are obtained by running the models through this test dataset.



(a) SingleFive (b) PairTen (c) New Dataset

Figure 3.7: Hands datasets

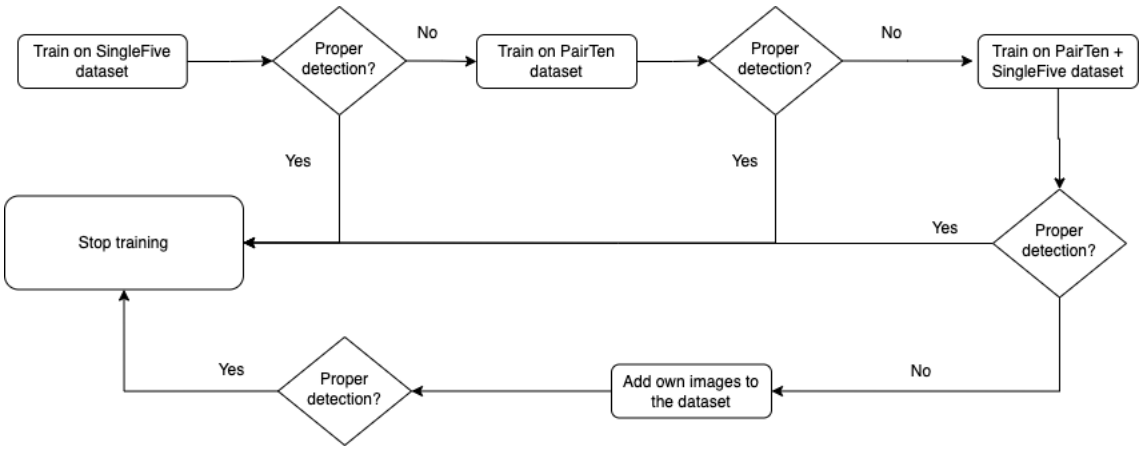


Figure 3.8: Dataset selection process

Dataset used	Thumb	Index	Middle	Ring	Little
EgoGesture SingleFive	7.17%	16.59%	12.28%	19.17%	37.86%
EgoGesture PairTen	28.48%	30.27%	32.08%	54.12%	83.29%
EgoGesture SingleFive + PairTen	59.29%	78.54%	25.63%	47.29%	**94.05%**
SingleFive + PairTen + TAURIS dataset	**84.10%**	**92.45%**	**79.37%**	**99.09%**	85.10%

Table 3.3: Tiny-YOLOv3 model evaluation results

Gamma value	Thumb (%)	Index(%)	Middle(%)	Ring(%)	Little(%)
Original (∼600 lux)	84.1	92.4	79.4	99.1	85.1
80% (∼500 lux)	84.6	89.7	74.9	98.9	87.0
60% (∼380 lux)	78.2	78.3	68.4	87.3	81.9
40% (∼220 lux)	61.2	55.5	64.6	64.9	83.2
20% (∼40 lux)	0.1	0	3.5	1.4	4.5

Table 3.4: Fingertip detection accuracy under different lighting conditions

3.2.3 Detection under varying lighting conditions

While the primary focus of this research lies in fingertip detection, comparing the developed system with existing hand gesture recognition systems, particularly concerning their performance under varying lighting conditions, offers valuable insights. This comparison not only sheds light on the broader field of gesture-based interaction but also highlights potential areas for future exploration.

To assess the robustness of the TAURIS fingertip detection model under diverse lighting conditions, the brightness of images within the test set was systematically controlled using the gamma correction algorithm. This algorithm allows for the adjustment of image brightness through the following formula:

$$O_{image} = 255 * (\frac{I_{image}}{255})^{\frac{1}{\gamma}}$$

where:

O_{image} = = output pixel value [0, 255]
I_{image} = = input pixel value [0, 255].
γ = = gamma value

Figure 3.9 illustrates the samples of the images produced. Original images were collected under 600 lux illumination. The ideal lighting level requirement for schools in the UK is 500[3] lux and higher. Whereas, 300 lux is considered to be an acceptable illumination. This methodology aligns with previous research that utilised gamma correction for data augmentation and model testing (Casado-Garcıa and Heras, 2020; Kachouane et al., 2012; Galdran et al., 2017). Table 3.4 presents the results of the conducted tests.

Several studies have explored hand gesture recognition techniques with varying degrees of success in handling lighting fluctuations:

[3]https://www.lyco.co.uk/advice/lighting-for-schools-colleges-and-universities

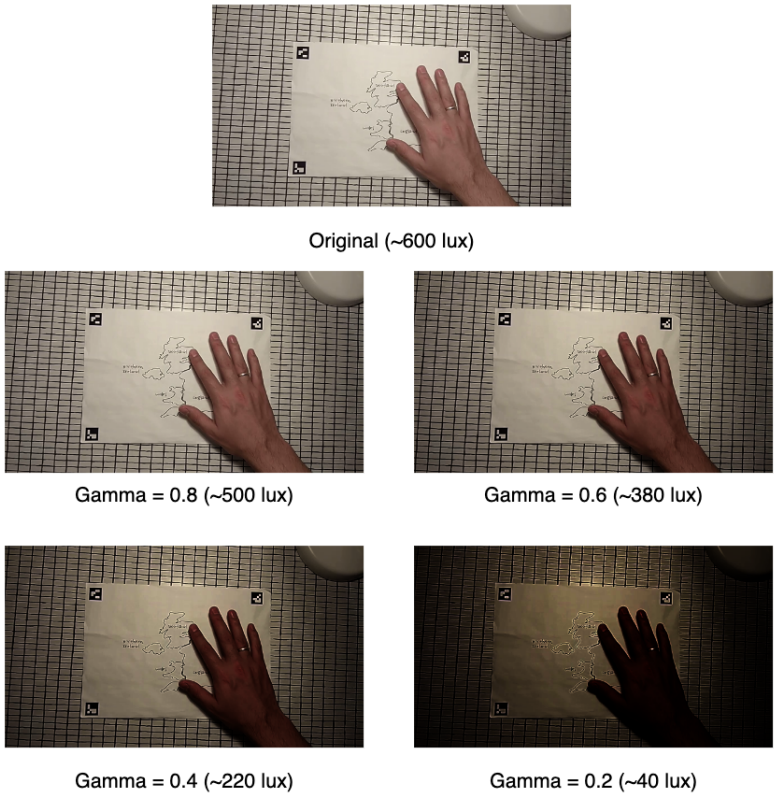


Figure 3.9: Image under different lighting conditions

- Wijayawardana (2021) employed OpenCV and a pre-trained model for hand gesture recognition to control a wheelchair, achieving recognition rates of 0%, 75%, and 100% at 250, 350, and 450 lux, respectively.

- Verdadero, Martinez-Ojeda, and Cruz (2018) utilized an AI model for static hand gesture recognition to control home appliances. While achieving a 100% recognition rate on a Samsung Galaxy S4 under controlled lighting conditions (minimum 120 lux), the authors emphasize the need for adequate illumination and contrast, suggesting potential limitations in real-world scenarios.

- Exploring the use of ambient light and photodiodes, Duan et al. (2020) leveraged an RNN to process data from an 8-photodiode array, achieving an impressive 99.31% accuracy for seven predefined hand gestures. However, a minimum illuminance level of 200 lux was required, and the potential impact of lighting changes on performance was acknowledged.

- Venkatnarayan and Shahzad (2018) introduced LiGest, a system utilizing a grid of light sensors to recognize gestures based on shadow patterns. Incorporating denoising, standardization, and rasterization techniques, LiGest achieved an average accuracy of 96.36% across diverse lighting conditions and user positions. However, the system's reliance on up to 6 light sources contrasts with the single light source employed in the TAURIS system,

Gamma value	200 lux	500 lux	600 lux
TAURIS (average for 5 fingers)	65.9	87.02	88.1
Wijayawardana (2021)	0	75	100
Verdadero, Martinez-Ojeda, and Cruz (2018)	-	-	100
Duan et al. (2020)	99.31	-	-
Venkatnarayan and Shahzad (2018)	65	-	96.36
Li et al. (2018)	-	-	96.36

Table 3.5: Systems detection accuracy under different lighting conditions

highlighting potential challenges in low-light environments.

- Li et al. (2018) presented a self-powered gesture recognition module utilizing photodiodes for both energy harvesting and sensing. Their system, based on a CFAR algorithm, demonstrated robustness against ambient light fluctuations, achieving high accuracy (above 96%) in normal office conditions (600 lux).

Table 3.5 offers a comparative evaluation of the TAURIS system's accuracy alongside other gesture recognition systems under varying illumination levels. Despite the inherent challenge of differentiating fingertips due to their similar appearance, the TAURIS system shows acceptable robustness, maintaining relatively high accuracy even at lower lux levels compared to some systems designed for pre-defined hand gestures. However, it is crucial to acknowledge that direct comparison between these systems is limited due to the differences in methodologies and tasks employed in each study. Further investigation is necessary to comprehensively understand the impact of diverse lighting conditions on the performance of various gesture recognition systems, particularly those focused on the intricate task of fingertip detection.

3.2.4 Detection in complex scenarios

To further assess the robustness and generalisability of the fingertip detection model, a new dataset, called the "Complex Scenarios", was created. This dataset consisted of 200 images captured challenging real-world conditions that might impact fingertip detection accuracy. These conditions included:

- **Occlusions:** A significant source of occlusions in the dataset were primarily due to the natural interaction of the two hands, where fingers from the right and left hands frequently overlapped or partially obscured each other. While occlusions by external objects like books or pencils were also considered, the emphasis was placed on capturing the hand-to-hand occlusions that are particularly prevalent during two-handed tactile graphic exploration.

Thumb	Index	Middle	Ring	Little
55.12%	60.34%	58.13%	63.01%	50.57%

Table 3.6: Fingertip detection in complex scenarios

- **Shadows:** In addition to the variations in illumination intensity investigated in Section 3.3.3, this dataset specifically incorporated images captured under lighting conditions designed to cast distinct shadows across the hands. The aim was to assess the model's robustness to the presence of shadows, which can obscure fingertip features and potentially lead to inaccurate detections.

- **Diverse Hand Shapes and Sizes:** The dataset included images of hands from individuals of different ages and hand sizes to capture a wider range of potential variations.

The Complex Scenarios dataset was used to evaluate the performance of the Tiny-YOLOv3 model trained on the original EgoGesture and TAURIS datasets. Table 3.6 presents a breakdown of the model's performance on the Complex Scenarios dataset for each fingertip class.

While the model's overall performance on the Complex Scenarios dataset was satisfactory, the results highlight the need for further improvement, especially in handling occlusions. The challenges posed by occlusions, particularly those arising from the interaction of the two hands, highlight the need for robust techniques to mitigate their impact on fingertip detection accuracy. Data augmentation, a widely used strategy in deep learning, offers a potential solution by artificially introducing occlusions into the training images, thereby enhancing the model's ability to generalise to such scenarios. Additionally, the implementation of a Kalman filter, which will be discussed in detail in the subsequent section, can further improve fingertip tracking stability by leveraging temporal information to predict and smooth fingertip trajectories.

3.2.5 Algorithm improvement

All or nothing

Even though a comparatively high detection rate for each fingertip was achieved, there still were some cases when the model confused the fingers and which were affecting the overall performance of the app. This was primarily due to similar appearances between fingertips. In order to minimise the number of such instances, a modified algorithm was implemented. I called it the "all or nothing" algorithm, i.e the coordinates of the index finger (in our case) were returned when all of the fingers, except the thumb, were detected. The thumb was neglected because its appearance differs a lot from the rest of the fingers and the detector almost never detected it as a different finger. The piece of pseudocode which describes the algorithm is presented below:

1: ***IF*** *(indexDetected == True && middleDetected == True && ringDetected == True*

```
2:      && littleDetected == True) THEN
3:          return indexPosition
4:      ELSE
5:          do nothing
```

Median filter

In addition, there was one issue that was encountered when the user was exploring TG with both hands- the detection algorithm was identifying the left-hand index finger instead of the right one. The number of such cases was minimal because the model was trained on a dataset that contained images of both hands. However, it had a negative impact on the performance of the entire application. To address this problem, a median filter was applied to smooth the input data. As a result, even if there was a case of misdetection, the algorithm was able to filter it out. After running the final model on a test set, it was found that the number of false detections for the index finger was less than 5%. This meant that the model was incorrectly detecting a fingertip in one of the 20 frames. Therefore, the window size of 5 consecutive values was more than enough to ensure that those detections were removed. The implemented algorithm is presented below:

```
1:      IF indexPosition size >= 5 THEN
2:          FOR indexPosition[i] from 0 to 5 DO
4:              window[] += indexPosition[i]
5:              i = i + 1
6:          sort entries in window[]
7:          median = window[3]
8:      ELSE
9:          do nothing
```

Kalman filter

To further enhance the robustness and stability of fingertip tracking, a Kalman filter was integrated into the TAURIS system. This widely used algorithm leverages temporal information from consecutive video frames to predict the future location of a fingertip, effectively smoothing its trajectory and mitigating the impact of spurious or missing detections. The Kalman filter operates through a two-step process of prediction, based on the fingertip's previous state and a motion model, followed by an update that incorporates the current measurement from the YOLOv3 detector to refine the prediction. See Appendix C for the details.

To evaluate the impact of the Kalman filter, a test was conducted using 100 frames of video footage. The initial detection model correctly identified the index finger in **89 out of 100** frames. After incorporating the Kalman filter, the accuracy improved to **95 out of 100** frames, demonstrating the filter's effectiveness in enhancing detection stability.

The computational load introduced by the Kalman filter is generally negligible compared to the processing time required for the YOLOv3 object detection. The filter's calculations primarily involve matrix operations on relatively small matrices, and modern mobile devices are capable of executing these operations efficiently. The addition of the Kalman filter did not impact the overall inference speed of the system, ensuring that real-time performance is maintained.

Figure 3.10 illustrates the whole fingertip detection process. The described "all-or-nothing", median filter and Kalman Filter algorithms are highlighted in green.

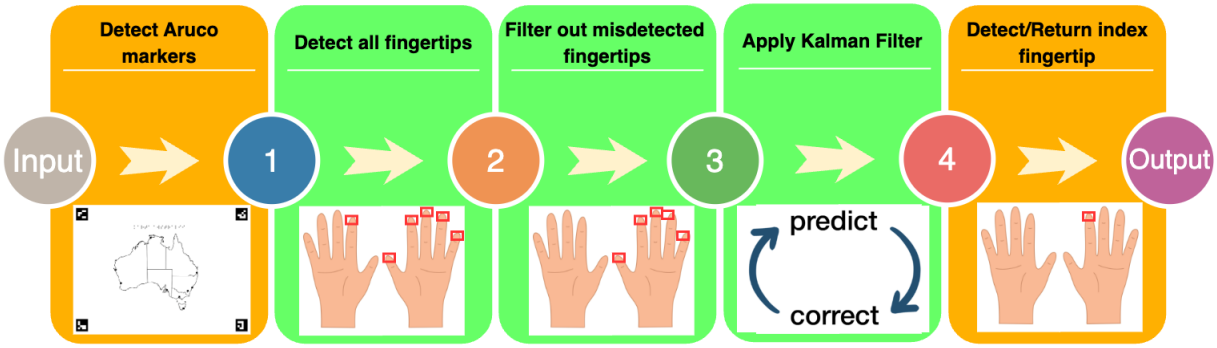


Figure 3.10: Fingertip detection algorithm overview

3.3 Conclusion

In this chapter, I first provided a technical background on developing CV algorithms. The purpose of this section was to introduce the fundamental concepts that were necessary for fingertip detection in my research. Accurate and fast fingertip detection is a very important part of the functionality of the developed app. In fact, this was also the most challenging part to achieve. To the best of my knowledge, there is no system that was implemented on a mobile device capable of simultaneously detecting the fingertips of both hands in real-time and under different lighting conditions. This chapter has provided a deeper insight into the procedures which were introduced to develop a better (more accurate and faster) detection model. After comprehensively evaluating the most suitable state-of-the-art object detection models, a tiny version of YOLOv3 was selected. As expected, this DL model architecture has shown poor accuracy performance - around 22% at the beginning (opposing to 57% and 50% achieved by the competitors). However, its detection speed was exceptional - 2.13 ms to process a single frame (vs 24.3 ms and 33.8 ms in the other two). Real-time execution was essential for the developed app. Thus, it was decided to carry on the research with a tiny-YOLOv3 and try to improve its detection accuracy. After enhancing a training dataset with more images, the model detection accuracy increased to 92% (for the index finger). On top of this, the "all-or-nothing", median filter and Kalman Filter algorithms were applied to minimise the number of wrong detections and smoother tracking. All evaluations were implemented in the same testing set and using the same hardware. Overall, it

can be concluded that the researcher achieved the goal of creating a robust model which allowed him to carry out the experiments and gather meaningful results.

While this research primarily focuses on fingertip detection, a comparative analysis with existing hand gesture recognition systems, particularly concerning their performance under varying lighting conditions, offers valuable insights. The TAURIS system, employing a customized Tiny-YOLOv3 deep learning model, demonstrates high accuracy for fingertip detection under ideal lighting conditions (600 lux) but experiences performance degradation as illumination diminishes. Alternatively, hand gesture recognition systems utilizing OpenCV often require controlled lighting and high contrast for optimal accuracy. Approaches leveraging ambient light and photodiode arrays, along with self-powered systems employing algorithms like CFAR, present promising avenues for handling diverse lighting conditions. Furthermore, shadow-based recognition systems, such as LiGest, exhibit robustness against illumination changes by interpreting shadow patterns created by hand movements.

Chapter 4

TAURIS System

4.1 Introduction

The main purpose of the TAURIS system is to allow Visually Impaired People (VIP) to explore tactile images without needing the help of a sighted person. The system consists of three main components: an Android mobile app, a Tactile Graphics (TG) online annotation tool, and pre-labelled TG. The mobile app is specifically designed for VIPs to use, allowing them to navigate and interact with tactile images on their own. On the other hand, the TG annotation tool is intended for teachers and instructors to use, enabling them to add labels and annotations to the tactile images. In this way, the TAURIS system aims to provide VIPs with a self-sufficient means of accessing and understanding tactile images, whilst also providing a means for educators to enhance the learning experience for visually impaired students. A thorough description of these instruments is presented in this chapter. The chapter[1] starts by discussing the phone app in Section 4.2, including its overview and the CV algorithms it uses. Section 4.3 then focuses on the description of the TG annotation tool and design requirements for TG production. Finally, a summary of the entire chapter can be found in Section 4.4.

4.2 Mobile Application

4.2.1 Overview

The main purpose of the developed app is to give users information about TG. It does this by tracking the user's finger positions as they explore the TG and providing information on what they touch in real-time. The previous chapter provided a detailed explanation of the fingertip detection process. This section will discuss the processes that take place before the detection step. The flowchart with the processing steps of the app can be found in Figure 4.1. The App

[1]Some of the work in this chapter has appeared in Zeinullin and Hersh (2022). Maralbek Zeinullin is the first author and main contributor to this paper.

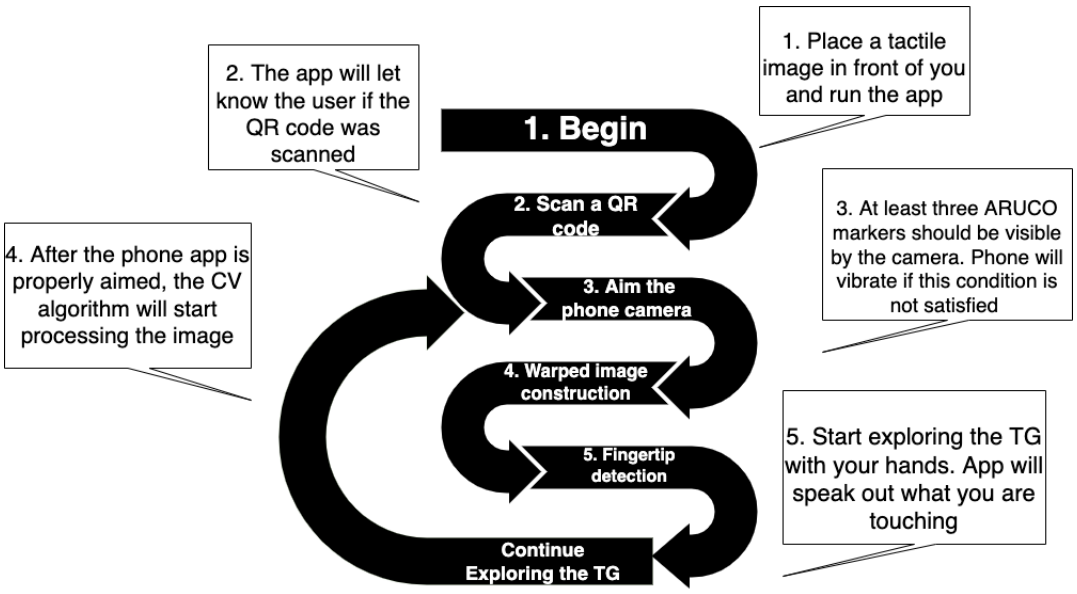


Figure 4.1: App working algorithm

was installed and tested on a Samsung Galaxy A52 device running the Android 11 operating system. This mobile device was used during the experiments. A capable camera of the device, a good chipset driver, and most importantly a moderate selling price (under £300[2]) were the main criteria for the researchers. It was crucial to test the application on a mobile phone that would be affordable for all potential users.

First, a QR code located at the back of the printed TG must be scanned (Figure 4.2). Using this method, the system will download the information about the TG to the device's memory. After users receive a notification that the code is successfully scanned, they can turn over the TG and start exploring it. The TG were printed on A4 ZYTEX2 swell paper. To explore TG with both hands, a phone holder, as shown in Figure 4.3, could be used. First, the app will look for the square markers located at the corners of the TG and will proceed if at least three of them are in the camera view of the phone. These markers are required to construct a "birds-eye" view of the image (Figure 4.4). Using this method, the app continuously updates the input image and calibrates it if the phone and/or TG positions are changed. The markers are called ARUCO and will be briefly discussed in the following section of this chapter. After markers are detected, the app will summarise the content of the visual information and its key features. TG is divided into 2400 (60x40) cells, each of which contains a predefined piece of information (see Section 4.2.4). As the Deep Learning (DL) model detects all visible fingers, it outputs the position of the default finger, which has been initially set as the index finger (this can be changed in the app's settings). Finally, the location of the fingertip is mapped onto the image cells and the data corresponding to it is read out to the user. It should be noted that the app uses both square QR codes and square ARUCO markers but their functions are different. It utilises a QR code to link

[2]https://www.pricerunner.com/Mobile-Phones/Samsung-Galaxy-A52-128GB-Compare-Prices

Figure 4.2: QR code printed on the back of the tactile graphics

Figure 4.3: Phone mounted on a special holder

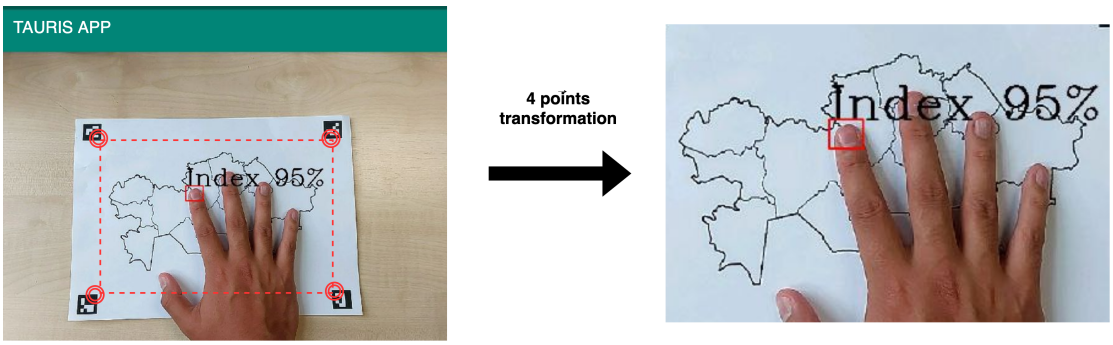


Figure 4.4: Four points birds-eye view image transformation
Note: Corners used to construct a new image pointed out inside red circles. **Index 95%** means that the app is 95% confident that the detected object is an index finger

to the data to be downloaded from the server and ARUCO markers to determine the location of the image. Information about these two algorithms will be presented in Sections 4.2.2 and 4.2.3 respectively.

On average, the app is capable of processing ∼15 frames per second[3] but it will not trigger feedback on every instance. Once the finger is moved, the system will play an audio output and wait for it to change location again. If the location does not change significantly or remains within the single TG area it will remain silent. To hear the audio output again, the user must remove their hand from the camera's view and then place it back in the TG area of interest. One of the advantages of the developed DL algorithm is its high accuracy in detecting each finger. This was discussed in more detail in the previous chapter in Section 3.2.2. Therefore, the user can explore an image with the whole hand and an algorithm will return the position of the desired finger only. This approach was chosen to avoid confusion because the size of a user's whole palm is quite large and the user might be covering multiple objects in one instance. Whereas the area of a single fingertip is much smaller and the users know what exactly they are touching at a given moment. In other studies, it is required to point with the index finger only (Baker et al., 2014; Fusco and Morash, 2015; Reichinger et al., 2016), OrCam[4], which is inconvenient, as discussed in the previous chapter. I will present the tools and algorithms that are used to ensure the app's proper functionality in the following sections.

4.2.2 QR code

QR (Quick Response) codes are specially designed two-dimensional arrays of black and white squares (Figure 4.5). The information stored in the codes can be easily read by QR scanner or a smartphone camera. Similar to Braille legends, text encoded into the QR code can be placed near the TG element. The advantage of the QR label is the amount of data it is capable of storing. According to the online source, it can handle up to 4296 alphanumeric characters (QR-

[3]Android FpsMeter function was used to measure the app image processing rate
[4]http://www.orcam.com

Figure 4.5: QR code sample

code-generator.com, 2020). This number varies depending on the size of the QR code array. The study conducted by Baker et al. (2014) reports that QR codes store up to 45% more text than Braille legends of the same size. The limitation of this method is the process of proper camera aiming to scan the code. Vázquez and Steinfeld (2012) analysed various approaches to mitigate this issue. According to the results, VIP preferred speech instructions for accurate camera focalisation. In my work, the QR code is used to download predefined information to the phone app. When the app starts running, it instructs the user to scan the QR code that has already been generated and placed on the other side of the TG automatically by the web tool. The QR code contains the URL link to the cloud server where the information about TG is stored. To reduce friction for the user when scanning, the generated QR code has a larger size. Finally, the app notifies the user if the code was scanned successfully so the exploration process can be initiated. To sum up, QR codes can be used as a supplementary tool for TG labelling, provided that the user is comfortable with code scanning.

4.2.3 ARUCO Markers

ARUCO is an open-source library that is used to detect square markers (Figure 4.6). The main advantage of the ARUCO is its fast and robust marker detection. The experimental results show that this algorithm works faster compared to other marker detectors maintaining the same accuracy (Romero-Ramirez, Muñoz-Salinas, and Medina-Carnicer, 2018). After successful marker detection, an algorithm returns a marker ID number with four corner coordinates. The library is based on OpenCV (Bradski and Kaehler, 2008) and its code is written in C++. In the context of the developed system, I use markers with IDs: 0, 1, 2, and 3 which are placed in clockwise order starting from the top left. First, the coordinates of each marker corner that is closest to the centre are detected. In total, sixteen corners are detected (4 corners x 4 markers) but in this project I use only four of them. Figure 4.7 highlights the corners used in the detection algorithm. Since it is known which ID corresponds to each of the corners, it is easy for the app to detect if one of the corners is not visible and then to estimate its location by utilising the information about the remaining three points (Equations 4.1 - 4.4).

$$P_{top\_right_{xy}} = P_{top\_left_{xy}} + P_{bottom\_right_{xy}} - P_{bottom\_left_{xy}} \tag{4.1}$$

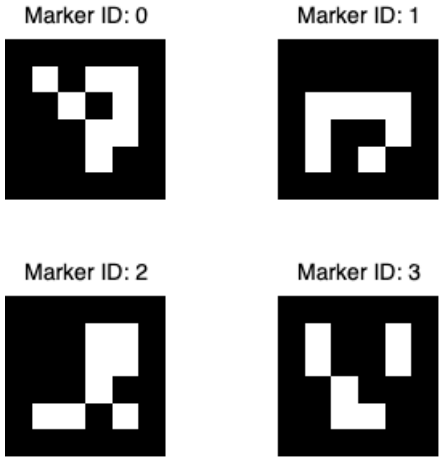


Figure 4.6: ARUCO markers

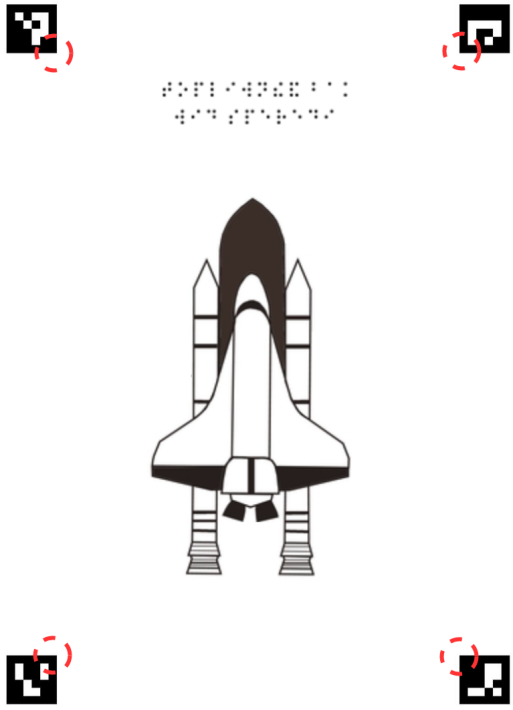


Figure 4.7: Tactile image with ARUCO markers

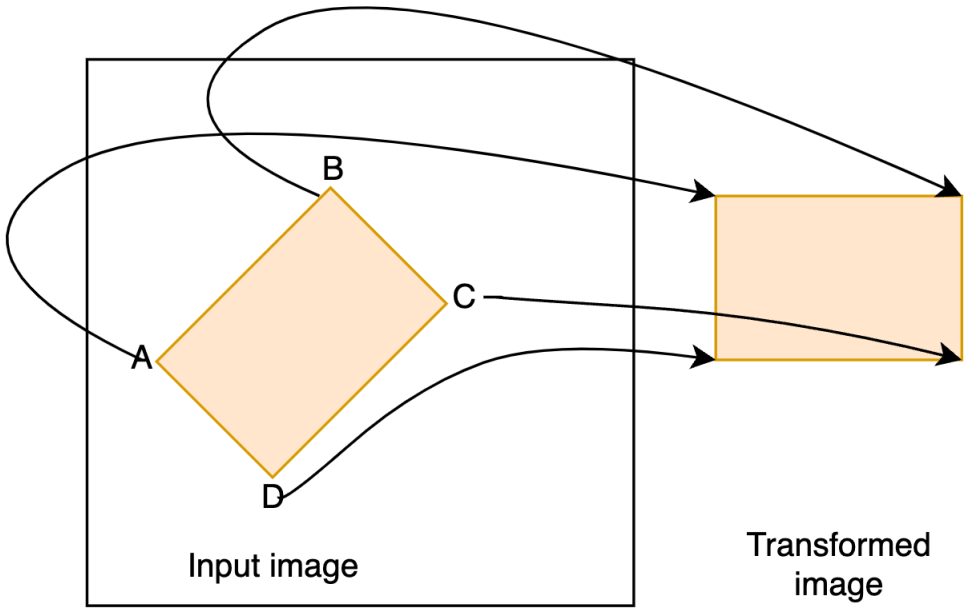


Figure 4.8: Input image perspective transformation

$$P_{bottom\_right_{xy}} = P_{top\_right_{xy}} + P_{bottom\_left_{xy}} - P_{top\_left_{xy}} \tag{4.2}$$

$$P_{bottom\_left_{xy}} = P_{bottom\_right_{xy}} + P_{top\_left_{xy}} - P_{top\_right_{xy}} \tag{4.3}$$

$$P_{top\_left_{xy}} = P_{bottom\_left_{xy}} + P_{top\_right_{xy}} - P_{bottom\_right_{xy}} \tag{4.4}$$

where:

$top-right, bottom-left, etc.$ = corner locations
xy = x and y coordinates of the corner point

Afterward, these four points are used to build a top-down view of the image. A top-down view is a type of perspective in which an image is depicted as if the viewer is looking down at the scene from above. One advantage of this process is that it only provides information about what is inside the rectangle of interest. Figure 4.8 illustrates how the algorithm focuses only on the area enclosed by the ARUCO markers. This process updates the input image in real-time, allowing the app to calibrate itself if the phone or TG orientation slightly changes. Figure 4.9 shows how the app deals with the situation when the TG orientation was changed. Since the app knows the coordinates of the four points in the source and destination images, a perspective transformation matrix can be easily determined using equation 4.6. To clarify, the four coordinate points of the source image are identified through the ARUCO markers. Four points of the destination image are selected, with the coordinates of the top left corner being

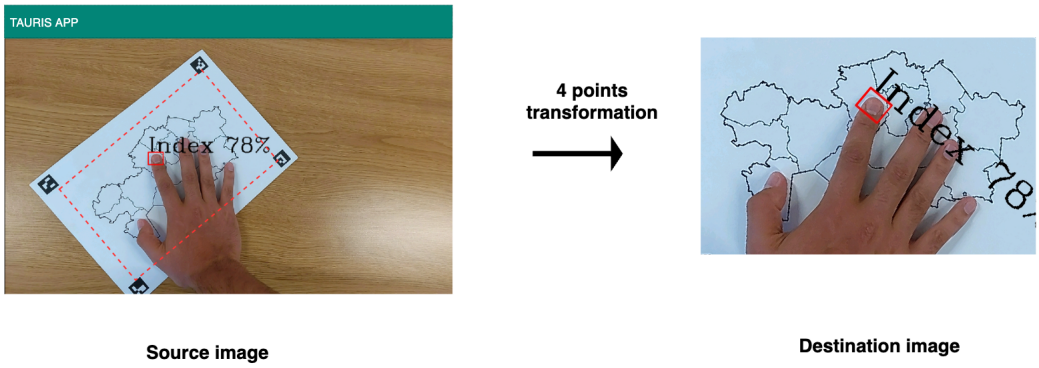


Figure 4.9: Example of input image perspective transformation

0,0 and the image dimensions being 900 by 600 pixels. Consequently, the coordinates of the detected finger will be translated accordingly.

After that, this matrix is used to transfer all enclosed points from the source image to the destination one using formula 4.5. OpenCV library was used to calculate a perspective transform matrix and then create a new image. It is worth highlighting that at least three markers should be in the field of view of the camera for the app to function properly.

Again, it is absolutely crucial to maintain consistent use of marker IDs. If markers with different IDs are used or their order is changed, the app will not be able to apply the perspective transform algorithm. By implementing these steps, the app will accurately detect the user's fingertip position (described in the previous chapter) with respect to the tactile image even if the phone orientation is continuously changing.

$$\begin{bmatrix} sx' \\ sy' \\ s \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ c_1 & c_2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4.5}$$

$$\begin{bmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ c_1 & c_2 & 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}^{-1} \begin{bmatrix} sx' \\ sy' \\ s \end{bmatrix} \tag{4.6}$$

where:

s = scaling factor
x', y' = transformed points (destination image points)
x, y = initial points (source image points)
a_1, a_2, a_3, a_4 = rotation, scaling matrix
b_1, b_2 = translation vector
c_1, c_2 = projection vector

Research work	**Camera aiming success rate (%)**	**Notes**
TAURIS	100	Based on user study with vibration alerts for missing markers.
Lee et al. (2019)	92	Based on user study with audio-haptic feedback.
Feiz et al. (2019)	89.5	Based on user study with audio instructions for form filling.

Table 4.1: Comparison of camera aiming techniques

The rotation matrix is used to rotate a point around a fixed origin and the scaling matrix is used to stretch or shrink the size of a geometric object along the x-axis and y-axis.

In this work, ARUCO markers help the users to accurately aim the phone camera. This involves adjusting the camera's position, composition, and angle to get the best possible image. As described above, the marker is placed at each of the four corners of the TG. Thus, the whole area of interest is enclosed by these markers. The phone notifies the user if, during the exploration process, more than two markers are not visible by making short vibrations. This helps the user to readjust the angle and position of the phone and to make sure that the whole image is in the field of camera view. This feature was acknowledged by the participants and was particularly helpful during the experimental study. Notably, all participants in the study successfully achieved proper camera aiming using this method. Since this approach requires preparing and printing the graphics with the markers in advance, it cannot be easily transferred to other systems and solve the problem of an accurate camera capturing real-time activities. In the future, an active navigation feature will be added. This feature will give explicit instructions to the user if the camera is not aimed correctly. Since the app knows the marker with which ID is missing, it can navigate the user and say that "top left" or "left side" (if two markers are not visible) is not in the field of view.

Table 4.1 presents a comparison of camera aiming success rates across different studies and systems. While direct comparison is limited due to variations in methodologies and tasks, the TAURIS approach, demonstrates a 100% success rate in achieving proper camera aiming within the context of its specific application and user study. This high success rate may be attributed to several factors, including the explicit spatial referencing provided by the markers and the active guidance through vibration alerts. However, it is important to acknowledge that the studies employed different performance metrics and involved participants with varying levels of experience and visual impairments, which can influence the observed outcomes.

The challenge of accurate camera aiming remains a persistent issue for visually impaired users of mobile technology. While various approaches have been explored, the ARUCO marker-based system employed in the TAURIS app demonstrates promising results, achieving a 100%

success rate in guiding participants towards proper camera alignment within the context of the study. The system's effectiveness can be attributed to its explicit spatial referencing, active guidance mechanisms, and combined use of visual and haptic modalities. Further research and development in this area hold the potential to enhance accessibility and empower visually impaired individuals to fully utilise the capabilities of mobile camera technology in their daily lives.

4.2.4 Mapping algorithm

The final rectangular image is divided into 2400 cells, each cell having a unique ID and the description associated with it. Whenever the user touches a particular cell, a linked predefined audio feedback is triggered. The destination image has dimensions of 900 pixels in width and 600 pixels in height. Each cell in the grid has dimensions of 15x15 pixels. This means that the grid has 60 cells horizontally and 40 cells vertically, resulting in a total of 60 * 40 = 2400 cells.

The specific design choices for the grid size and number of cells are based on both the physical dimensions of the TG and ergonomic considerations. The TG enclosed by four ARUCO markers has an actual physical size of 24 × 16 cm, which translates to individual cell dimensions of 4 × 4 mm. This cell size is smaller than the average width of the human index fingertip, reported to vary between 16 and 20 mm (Dandekar, Raju, and Srinivasan, 2003). Additionally, Johnson and Blackstone (2007) found the mean width of the index fingertip to be 20.3 ± 2.4 mm across participants. The smaller cell size ensures precise mapping and avoids ambiguity when the fingertip interacts with the TG. Furthermore, research suggests that many visually impaired people prefer using their index finger as the leading one for tactile exploration (Wong, Gnanakumaran, and Goldreich, 2011).

The cell IDs are assigned sequentially starting from the top left corner, with the first cell having an ID of 1 and the last cell having an ID of 2400. The cells are numbered by going from left to right, then top to bottom. After the fingertip is detected, its coordinates are linked to the individual cell. Equation 4.7 is used to associate the location of the fingertip with the cell ID. Note that when the coordinate values of the fingertip bounding box are divided by the cell dimensions, the result is rounded up to the closest integer. For instance, if the fingertip is located right in the middle of the image, its coordinates are c_x, c_y = 450, 300. Assuming that the image width is 900 pixels and the dimensions of an individual cell are 15x15, the corresponding cell number can be easily calculated. After applying all values to the equation, the corresponding cell ID is equal to 1170. Figure 4.10 demonstrates how the fingertip coordinates are mapped onto the individual cell. The algorithm uses the centre coordinates of the bounding box and then associates it with the cell closest to it. In the figure, the target cell has a red circle inside.

$$C = \frac{W_{image}}{W_{cell}}(\lceil \frac{c_y}{W_{cell}} \rceil - 1) + \lceil \frac{c_x}{H_{cell}} \rceil \tag{4.7}$$

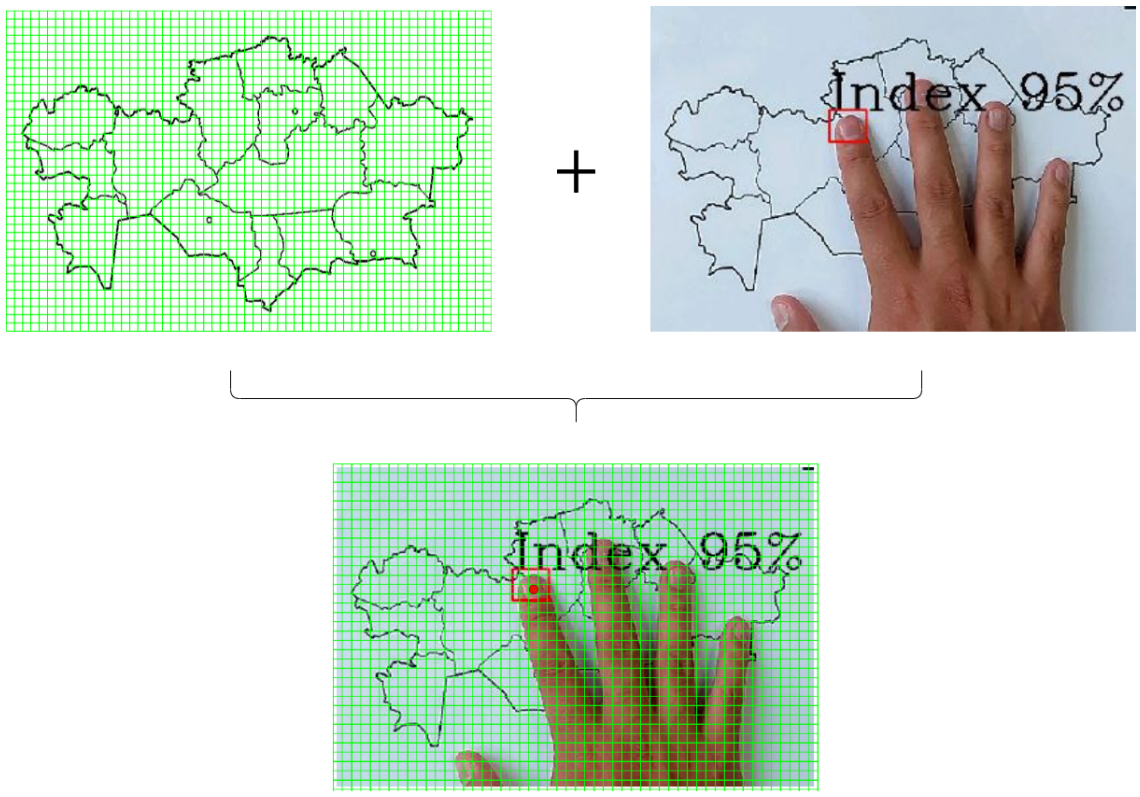


Figure 4.10: Fingertip location mapped on the image grid cells

where:

C = Cell ID
W_{image} = Image width
W_{cell} = Cell width
H_{cell} = Cell height
c_x, c_y = x and y coordinates of the centre of the bounding box

4.2.5 Description modes

Depending on the type of information users need, different descriptions can be requested. There are three types of descriptions available: overview, basic, and detailed.

- Once the QR code is scanned and the TG is turned over, the **overview** is activated. This provides general information about the content of a TG. Turning over the graphics in both directions again will allow the user to hear this information one more time.
- The app provides a **basic** description as soon as the user begins exploring the TG and the algorithm detects their fingers. When the app is in this mode, it tells the user which object their index finger is touching without providing any further information.

- The **detailed** description mode can be activated by the user if they would like more detailed information. This can be achieved by holding the index finger still for three seconds above the target object.

To clarify the differences between what type of information is presented in each mode, examples of descriptions for the map of Australia are provided below.

- **Overview**: This is the map of Australia. It consists of six states and two territories.

- **Basic**: New South Wales

- **Detailed**: Capital city of this state is Sydney. Australian Capital territory is also located in this state.

To obtain a quality description, it is essential to provide accurate and concise information. Long text can exhaust the user, while irrelevant information can confuse them and waste their time. It is also crucial to provide the most important information. "How to Write Alt Text and Image Descriptions for the Visually Impaired"[5] created by the Perkins School for the Blind is a useful guide.

4.2.6 Implementing Kazakh TTS

To improve the accessibility of the TAURIS app for use in Kazakhstani schools for the blind, it was necessary to incorporate a Kazakh language text-to-speech (TTS) synthesiser. According to the latest census data[6] in Kazakhstan, over 80% of the population understands the Kazakh language. Additionally, during the interview phase, several participants highlighted the need for a Kazakh language option (See Chapter 6). Given the absence of a native Kazakh language TTS on Android devices, the integration of a third-party API was required. The Scientific and Practical Center named after Sh. Shayakhmetov (2024) “Til-Qazyna” released a Kazakh TTS recently and this synthesiser was successfully integrated into the TAURIS app through their API. I conducted preliminary testing of the TTS and it has demonstrated stable performance. However, further testing with end-users is required to comprehensively evaluate its effectiveness.

4.2.7 Computational Demands of Real-time Fingertip Detection

The implementation of real-time fingertip detection on a mobile device presents inherent challenges due to the computational resources required for neural network inference. This section

[5]https://www.perkinselearning.org/technology/blog/how-write-alt-text-and-image-descriptions-visually-impaired

[6]https://stat.gov.kz/ru/national/2021/

presents the performance characteristics of the TAURIS app, specifically focusing on its resource utilisation. The profiling tests were conducted on a Samsung Galaxy A52 device equipped with 4GB of RAM and a 4500 mAh battery, providing a representative evaluation of the app's performance on a mid-range smartphone.

Profiling Results and Analysis

The TAURIS app was profiled using Android Studio to analyse its resource consumption. The profiling data encompassed CPU usage, memory allocation, and battery consumption during the fingertip detection process.

- **CPU Usage:** The average CPU utilisation during continuous fingertip detection was measured at 37%, with peak usage reaching 61%. This moderate level of CPU utilisation suggests efficient use of processing power, allowing for concurrent execution of other app components and background processes without significant performance degradation.Although the chosen tiny YOLO lightweight model architecture prioritises efficient inference, the computational complexity inherent in deep learning models can still introduce delays, particularly on less powerful mobile devices.
- **Memory Usage:** Memory allocation exhibited an average of 30 MB, while peak usage reached 245 MB. This encompasses the memory footprint of the neural network model itself, as well as other app components and data structures involved in image processing and audio output generation.
- **Battery Usage:** Battery consumption was measured at 0.3% per minute during active fingertip detection. This relatively low energy consumption ensures prolonged usage of the TAURIS app without causing excessive battery drain, a crucial consideration for mobile accessibility tools.

Table 4.2 provides a comparative analysis of TAURIS's resource consumption against the performance benchmarks of WhatsApp during a video call and YouTube during a live stream on the same mobile device.

4.3 Web Tool for Annotations Creation

4.3.1 Overview

In addition to the phone application, an online tool to create annotations for the tactile images was developed. The website interface is presented in Figure 4.11. For now, this tool is designed for sighted users (parents, teachers, instructors, etc.) but in the future, the website might be adapted for the VIP as well. First, the user has to upload a sample of a tactile image in JPG

App	CPU Usage	Memory Usage	Battery Usage
TAURIS	37% (Average) 61% (Peak)	30 Mb (Average) 245 Mb (Peak)	0.3% per minute
WhatsApp	49% (Average) 73% (Peak)	79 Mb (Average) 251 Mb (Peak)	0.4% per minute
YouTube	42% (Average) 77% (Peak)	34 Mb (Average) 328 Mb (Peak)	0.2% per minute

Table 4.2: Comparison of TAURIS resource consumption against other Apps

Figure 4.11: TAURIS web tool

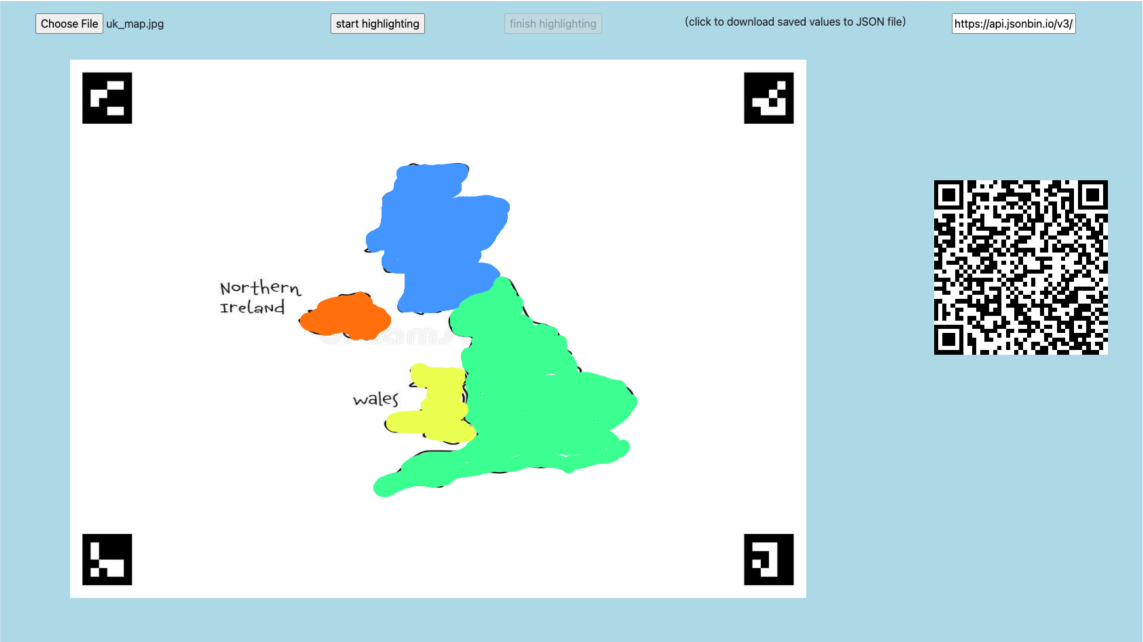


Figure 4.12: Image annotated by the TAURIS web tool.

or PNG formats. There are many online libraries that contain multiple sets of TG. The list of some online repositories can be found in Table 4.3. Instructors may need to create their own images if they are not available online. Next, the image needs to be divided into the regions of interest (ROI) and the corresponding descriptions should be prepared. After that, ROIs have to be highlighted and annotated with descriptions. The user can cancel the highlighting if a selection mistake was made. It is also possible to edit information about highlighted regions (Figure 4.13). Once all ROIs are marked, the user has to press the "finish" button for the tool to save all information to the server. At the same time, a PDF version of the tactile image will be created with the ARUCO marker frame. In addition, a QR code will automatically be generated and placed on the second page of the PDF file. Lastly, the final document can be downloaded to local memory and printed (double-sided mode). Figure 4.12 demonstrates the image annotated with the web tool. An automatically generated QR code is also shown.

4.3.2 Guidelines for Effective Tactile Graphic Design

Creating effective TG requires careful consideration of design principles that prioritize accessibility and clarity for the reader. The following key requirements, based on the guidelines established by the Braille Authority of North America (Miller et al., 2010), outline essential considerations for producing meaningful and usable TG. These principles emphasize conveying information efficiently, minimizing clutter, and adapting the design to the reader's specific needs and abilities. Following these guidelines ensures that tactile graphics serve as effective learning tools, providing equal access to information for Braille readers.

Figure 4.13: TAURIS web editing options

Link	**Resource description**
http://www.tactilelibrary.com/	Library of diagrams used in the education of the blind and with low vision
http://www.tactilegraphics.org/	A website with the information on designing and making TG created by Lucia Hasty
https://imagelibrary.aph.org/aphb/	Educational resources for visually impaired created by American Printing House for the blind
https://www.heardutchhere.net/	Manual on editing graphics for the blind
https://tactileimages.org/en/library/	Library of the TG created for the Tactile Images[7] app.

Table 4.3: Tactile Graphic Resources

- **Meaningful Representation:** Prioritize clear meaning over exact reproduction.
- **Effective Production:** Choose the best production method, not the cheapest.
- **Consistent Braille:** Maintain Braille code consistency with the main text.
- **Essential Elements:** Include only necessary visual elements.
- **Clear Layout:** Ensure a clear and uncluttered layout.
- **2D Preference:** Use 2D views unless depth is crucial.
- **Unified Dimensions:** Use consistent page dimensions for tactile graphics and Braille text.
- **Informative Notes:** Use transcriber's notes to explain changes and clarify content.
- **Reader Consideration:** Adapt design for the reader's age and skill level.
- **Orientation Cues:** Include tactile orientation cues on independent graphics.

4.4 Conclusion

One of the main aims of the research is to identify current gaps in the field of assistive technology for VIP and propose a solution that will help to fill those gaps and thus benefit the visually impaired community. To achieve this, a special educational system was designed. Limitations of the available methods were considered before the system development. The proposed system will meet present limitations in the following ways:

Labour-intensiveness: The special online web tool was developed that will allow the user to easily upload and label images. Furthermore, processed images can be saved on the website server so that other users can view and download them. Using this method, knowledge transfer and collaboration between teachers and instructors will be established. In the future, it is planned to add a feature that will allow users of the web tool to rate the created images, so the most successful ones can be easily recognised. In short, users will be able to contribute and utilise the best samples of TG which were created and assessed by the whole community.

Mental load on the reader: A developed system will accompany the TG with predefined audio descriptions. The previous study has emphasised the effectiveness of an audio-tactile approach (Melfi et al., 2020). Their experimental results show that users acquire information faster compared to other methods and show more than 70% accuracy while answering the questions. The authors have used an iPad that was programmed to trigger the audio output when a user touches the swell paper image placed on the tablet screen. In the phone app, I have implemented the same strategy but the technical execution is different. The app uses a phone camera to track fingertip locations in order to trigger the corresponding audio descriptions. Additionally, the

app can provide general information about graphics and give preliminary cues for the location of key elements. The conducted experiments also prove the effectiveness of this approach.

Production cost: Unfortunately, there is no way to reduce the production cost of the TG itself. Instructors have to use a regular swell paper and a fuser in order to produce a traditional tactile image. The reduction of these expenses and the optimisation of printing hardware is beyond the scope of my research. However, I propose a solution that decreases the labour effort for teachers and minimises the mental load for the students with an additional price of the smartphone with a camera only (which many users already possess). According to the survey (n=259, average age = 44.51), 95.4% of VIP use smartphones on a daily basis (Griffin-Shirley et al., 2017).

To sum up, a detailed description of the TAURIS system was presented in this chapter. The major component of the system is a mobile app. The app involves many features including ARUCO markers and QR code detection, image transformation and audio output generation. In addition, the application uses the object detection model described in Chapter 3. The web tool for TG annotations is a second component of TAURIS. General information about this online tool was provided in this chapter as well. In conclusion, the developed system differs from the existing solutions in several ways. First, as it is capable of detecting all ten fingers separately, it enables two-handed exploration for users. Second, the ARUCO markers placed at the corners make the camera aiming process more accessible. Third, a compact architecture of the neural network allows real-time execution even on a mid-range mobile device. To conclude, my system still relies on traditional TG production methods. But, on the other hand, it offers a solution that is advantageous in terms of ease of use and, in contrast to other assistive technology systems, requires a smartphone only.

Chapter 5

Methodology

5.1 Introduction

In this chapter[1], a methodological framework of the research is introduced. In section 5.2, the list of the research questions are presented. The appropriate research method is introduced and justified in this section as well. Next, in Section 5.3 the data collection procedures are described. Finally, the data analysis steps are discussed in section 5.4.

5.2 Research design

5.2.1 Research questions

This study investigates the usability and effectiveness of the TAURIS app as an educational tool for visually impaired individuals. Specifically focusing on its novel approach to Tactile Graphics (TG) exploration using a smartphone. This involved evaluating user experience, including the ease of camera aiming, and comparing the app's performance with traditional methods (Braille and screen readers) in terms of accuracy and speed of information access. The research questions guiding this investigation are:

1. *What are visually impaired individuals' **perceptions and attitudes** toward the use of smartphone app in the context of exploring Tactile Graphics (TG)?*

2. *To what extent does real-time speech output, integrated with tactile exploration, enhance the **comprehension and retention** of complex information conveyed through TG for visually impaired users?*

3. *What methods can be employed to improve **camera aiming** in smartphone-based assistive technology applications designed for exploring TG?*

[1]Some of the work in this chapter has appeared in Zeinullin and Hersh (2022). Maralbek Zeinullin is the first author and main contributor to this paper.

5.2.2 Mixed-methods design

Answering the research questions requires the collection of both quantitative and qualitative data. Therefore, a mixed-method research design was used. In this section, the advantages and disadvantages of this approach are reviewed.

In the context of my study, a major advantage of this method is not only the possibility to determine if the proposed system works better but also to understand how participants feel about using it. A mixed method approach is used to corroborate the findings and obtain thorough and unbiased results (Creswell, Plano Clark, et al., 2003). Additionally, combining the results of quantitative experiments with qualitative end-user interviews led to a more comprehensive understanding of the topic as well as a more extensive approach to the analysis (Denscombe, 2008). In my research, this methodological approach was used to evaluate the effectiveness of the proposed system and to obtain in-depth information on the perception of the end-user.

Furthermore, during this approach, a triangulation technique can be applied. This technique allows examining the same phenomenon by different methods. In our case, the experimental results showed how accurately the participants responded to the questions while using different exploration modes. Whereas a post-experimental session was conducted to gain a better understanding of how the novel method impacted the exploration process. Lastly, this method allows us to mitigate the impact of a small number of participants by gathering multiple types of data. Unfortunately, there are certain challenges associated with the use of this method as well. In particular, performing a mixed method analysis can be more difficult than either qualitative or quantitative analysis alone. This is because it requires combining data from both qualitative and quantitative sources and then interpreting them in the same context. To sum up, by introducing this approach the credibility, variation and quality of the data can be dramatically increased, given that the researcher knows how to effectively use this “mixture”.

5.2.3 Mixed-methods: The convergent design

There are three core designs of mixed methods research (Creswell and Clark, 2017):

1. **A convergent design** combines qualitative and quantitative data collection and analysis techniques to build a more comprehensive understanding of a research problem. It is often used to validate the findings of one method with those of another.

2. **The explanatory sequential design**. In this approach, quantitative data is collected and analysed first. Then qualitative data collection and analysis is performed to explain and expand the findings of the quantitative data.

3. **The exploratory sequential design.** The design involves the collection and analysis of qualitative data first, which is then followed by the collection and analysis of quantita-

tive data. This allows researchers to explore the research question in depth, while also gathering quantitative data to support their findings.

The main reason why a convergent design is often selected before explanatory and exploratory sequential designs is due to its ability to provide a comprehensive understanding of the research topic. The main goal of this approach is "to obtain different but complementary data on the same topic" (Morse, 1991). One advantage of this method is that it juxtaposes the strengths and weaknesses of the quantitative and qualitative approaches (Patton, 1990). This method is also named a concurrent triangulation where two databases (quantitative and qualitative) are used to corroborate findings and get a full understanding of a single topic (Creswell, Plano Clark, et al., 2003). After taking everything into account, it was decided that a convergent design is the most appropriate option for this study.

Following the procedures for carrying out a convergent mixed-method design, I performed the steps described below (Creswell and Clark, 2017). First, after obtaining ethical approval, structured interviews were conducted to gather participants' demographic information and data about their experience with TG & mobile devices. Next, quantitative measurements were obtained through a series of experiments. Subsequently, qualitative post-experimental interviews were carried out to record the participants' thoughts and opinions about the proposed system. In the final phase, teachers of visually impaired students were asked to provide feedback and insights on the TAURIS app, exploring its potential impact on pedagogical approaches and student comprehension. The quantitative and qualitative data from the initial phases and the teacher feedback were then analysed separately using the techniques discussed in Section 5.4. Finally, findings from all data sources were merged, compared, and interpreted in the *Discussion* section of this manuscript. Figure 5.1 summarises the described procedures.

5.3 Data collection methods and procedures

5.3.1 Participants

Initially, it was planned to carry out the experiments in one of the Glasgow schools. I have contacted multiple schools including the Hazelwood[2] but due to the COVID-19 situation at that time, the school administration advised to get back when the lockdown measures are eased. Therefore, I decided to gather data in my home country - Kazakhstan, where educational organisations were operating without major restrictions. Shymkent regional boarding school for visually impaired children agreed to take part in the study and permitted the investigator to conduct the experiments given that all safety and health measures are taken. This school provides educational services in a variety of subjects from the first (primary) to the eleventh (secondary) grade. The ethical approval to conduct this study was approved by the Ethics Committee of the

[2]Local School for children and young people with sensory impairment and complex learning needs

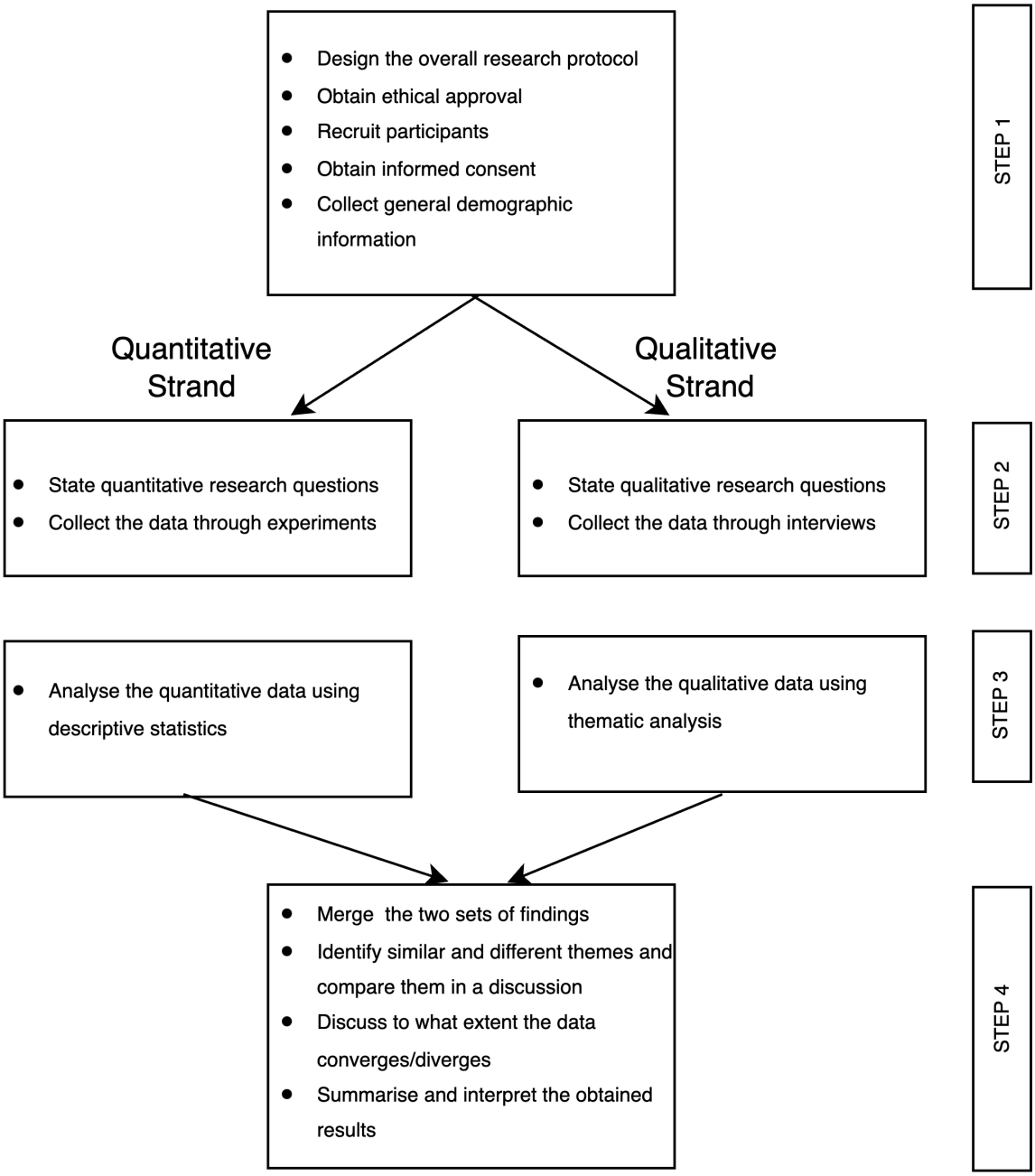


Figure 5.1: Convergent Mixed-Method Design Procedures

Glasgow College of Science and Engineering (Application Number: 300200167). In addition, the following ethical aspects were strictly considered:

1. Ensuring anonymity, privacy, confidentiality and compliance with data protection regulations
2. Assuring the ongoing consent
3. Assuring the safety of participants during experiments
4. Providing information in an accessible format

I recruited participants for my study with the help of the Shymkent school IT teacher, who recruited and coordinated with potential participants on my behalf. Since it was during the summer holidays, only a few students were available to participate, and the majority of the participants were visually impaired school library staff. The IT teacher was instrumental in helping me to reach out to and connect with these individuals. To further strengthen the study's findings and gather a more comprehensive dataset, an additional nine secondary school students and five school teachers were recruited and interviewed during the school term. Overall, the participant group was considered to effectively represent the main stakeholders relevant to the study.

All COVID-19 safety measures were taken into account during the experiments to ensure the health and well-being of all parties involved. This included wearing masks, using hand sanitisers, and thoroughly cleaning the experiment site after each session. By implementing these precautions, the transmission risk was minimised and a safe and productive environment was maintained for all participants.

The school library was chosen for conducting experimental sessions because it was a safe and familiar place for all visually impaired participants. They were able to navigate to the library safely and comfortably as they were already familiar with the layout and surroundings. This helped to ensure that the experiments could be conducted smoothly and without any unnecessary disruptions.

To ensure the session information sheet and consent forms were accessible, digital versions were sent to participants in advance. Before starting the experimental session, all consent forms were read out loud to the participants and signed by them. Some chose to use government-issued rubber stamps with their signatures to sign the documents. This helped to ensure that all participants were fully informed and gave their consent before the study began.

5.3.2 Pilot study

A pilot study was conducted with a single participant before carrying out the experiments to check whether all the materials were accessible and whether the design of the experiment was

feasible. As a result, three changes were made to the experimental protocol and the investigator obtained a good idea of session duration which facilitated scheduling subsequent sessions.

The main change was giving the participants time to familiarise themselves with the descriptions first, and only subsequently asking the question and starting to record the time for all three exploration modes. This was to avoid the response time for the first question being much longer than that for the remaining ones and the results not being normally distributed, as the pilot participant was found to require a significant amount of time to read the general TG descriptions in Braille.

Another insight that was revealed is the type of text-to-speech synthesiser (TTS) preferred by the participant. It turned out that most visually impaired users find the Google TTS more pleasant to the ear. With respect to this, the default Samsung TTS on the device was changed to Google TTS.

In addition, the pilot study gave the investigator an understanding of the duration of a single session. The pilot session took around 75 minutes to complete. This information was then used to schedule the following meetings. Lastly, it was found out that one of the Braille embossed texts was not fully readable, so it was replaced.

5.3.3 Phase 1 – Interview

As mentioned in 5.3.1, informed consent was obtained before the start of the study. The interviews were divided into two main parts. The first covered general demographic information and the second participants' experiences with the TG. This included questions about whether they had used TG in school and how it affected their learning. The final question asked about their familiarity with Assistive Technology (AT) applications for smartphones, including whether they had used apps that require the use of the phone's camera. The questions used are listed in Appendix B.

5.3.4 Phase 2 – Device testing (Quantitative)

Apparatus

The TAURIS app was installed and tested on a Samsung Galaxy A52 device running the Android 11 operating system. This mobile device was used in the experiments with all participants. The capable camera of the device, a decent chipset, and most importantly a moderate price (under £300[3]) were the main criteria for the research. It was crucial to test the application on a mobile phone that would be affordable to all potential users.

The TG were printed on A4 ZYTEX2 swell paper. Each TG had two versions: one with Braille labels and one without. The Braille labelled version was used by the participants during

[3]https://www.pricerunner.com/Mobile-Phones/Samsung-Galaxy-A52-128GB-Compare-Prices

the Braille text and screen reader modes sessions. The version without labels was used for the TAURIS App exploration mode. The quality of the TG and the Braille texts were assessed by the library staff before production to ensure that the printed graphics were readable. Participants were given up to ten minutes to familiarise themselves with the app before starting the actual testing session.

Experiments

The aim of the experimental session was to investigate how different modes affect the performance of the participants and therefore to answer research questions 2. Due to the small number of participants, a within-group design method was chosen to analyse the data. This method involves testing more than one treatment on the same participant. In our case, the same person was exploring TG with two different modes (app and either Braille or screen reader).

The independent and dependent variables in the experiments were:

- **Independent variables**: Mode (App and Braille / Screen reader)
- **Dependent variables**: Time and accuracy

My objective was to find out how different modes affected performance while exploring TG. One of the disadvantages of the within-group approach is the learning effect. Since each participant used the app three times in one session, he/she was gaining more experience throughout the experiment. Another issue associated with this method is fatigue. Each participant had to explore six different tactile images using the two modes. According to research, the appropriate length of one experiment session should be between 60 and 90 minutes (Nielsen, 2005). I tried to finish the sessions within the suggested time constraint and none of the sessions exceeded 90 minutes. Additionally, I examined how the participants' ages and level of vision loss affected their performance.

During the experimental session, each participant explored six different tactile images. Three tactile images were explored using the app and the remaining three with their choice of mode (Braille text or screen reader). Ten participants used Braille and another ten used a screen reader mode. The TG were divided into three categories: object, map and graph, with each category having a different type of associated graphics. Examples of TG used in the experiments are illustrated in Figure 5.2. Brief descriptions of the modes and the object and graphic types are provided below.

Exploration Modes

(1) App mode. The TAURIS app was used by participants in this mode. Before each test, I asked participants whether they prefer to explore TG with one or both hands. All of them preferred to use both hands to explore the images. Thus, the phone was mounted on a holder for all sessions. The app used a phone camera to track the user's fingertip locations in order to

trigger the corresponding audio description. Figure 5.3 shows an example of TG used in app exploration mode.

(2) Braille mode. In this mode, the embossed Braille text descriptions were provided with the TG. The user had to switch back and forth between the TG and the description sheet to read the description. For convenience, Braille legends were placed next to the objects illustrated on the TG (Figure 5.4).

(3) Screen reader mode. For this mode, text descriptions for the TG were printed on standard (rather than swell) paper. Participants used their own mobile phones and their preferred apps to capture the document and convert printed text to speech. Two participants used a standalone device[4] with optical character recognition (OCR) and audio output which converts printed text into speech since they did not have an appropriate document reader app installed. Response times were not affected by the choice of device, since all measurements were taken after the text was captured and spoken aloud once. Since the users who selected this mode were able to read Braille numbers (but were not fluent Braille readers), the TG with the Braille legends were used in this mode for their convenience (Figure 5.4).

Graphic Types

In addition, the TG were divided into three different categories according to the task type:

Object. The images of the frog lifecycle and the space shuttle were used in this part. First, participants had to listen/read the general information about the object presented on the TG. They then explored the object thoroughly and answered questions about it. This type of task was considered the most difficult due to the amount of information provided.

Graph. Two different histograms were used. The associated tasks were the easiest, as most of the information could have been acquired by touch, e.g. the length and location of the bar.

Map. The maps of Australia and Kyrgyzstan were used and the participants used the same exploration algorithms as described in the "Object" type.

Procedure

A tactile image was placed in front of the participant at the beginning of each task as shown in Figure 5.5. Then, a short summary of the image was presented in the appropriate exploration mode. In the app mode, the summary was provided in an audio format as soon as the QR code was scanned. A large QR code was placed on the other side of the paper at a point corresponding to the centre of the TG. The dimensions of the QR code in this study are much larger than in previous works. This makes it much easier for visually impaired users to locate and scan the code. Correspondingly, none of the participants experienced any difficulties in scanning the code during the experiments. In the screen reader mode, a text document with the summary was first scanned by the text reader app and then spoken out loud by the device. In Braille mode, a sheet with the corresponding Braille text was used.

[4]Standalone machine with a camera. Similar to this one https://www.visionaid.co.uk/standalone-reading-machines/readeasy-evolve

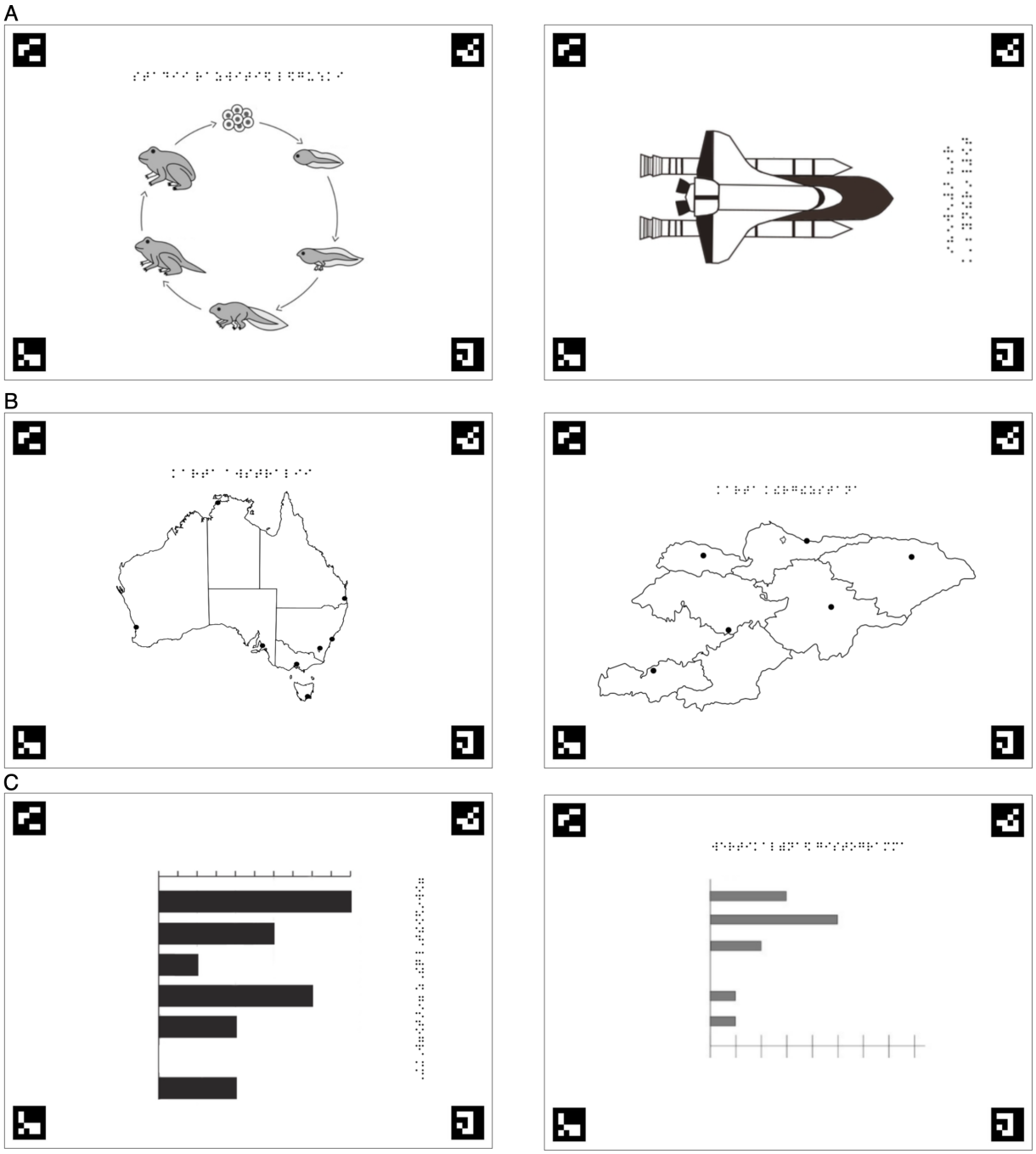


Figure 5.2: Examples of object (A), map (B) and graph (C) TG.

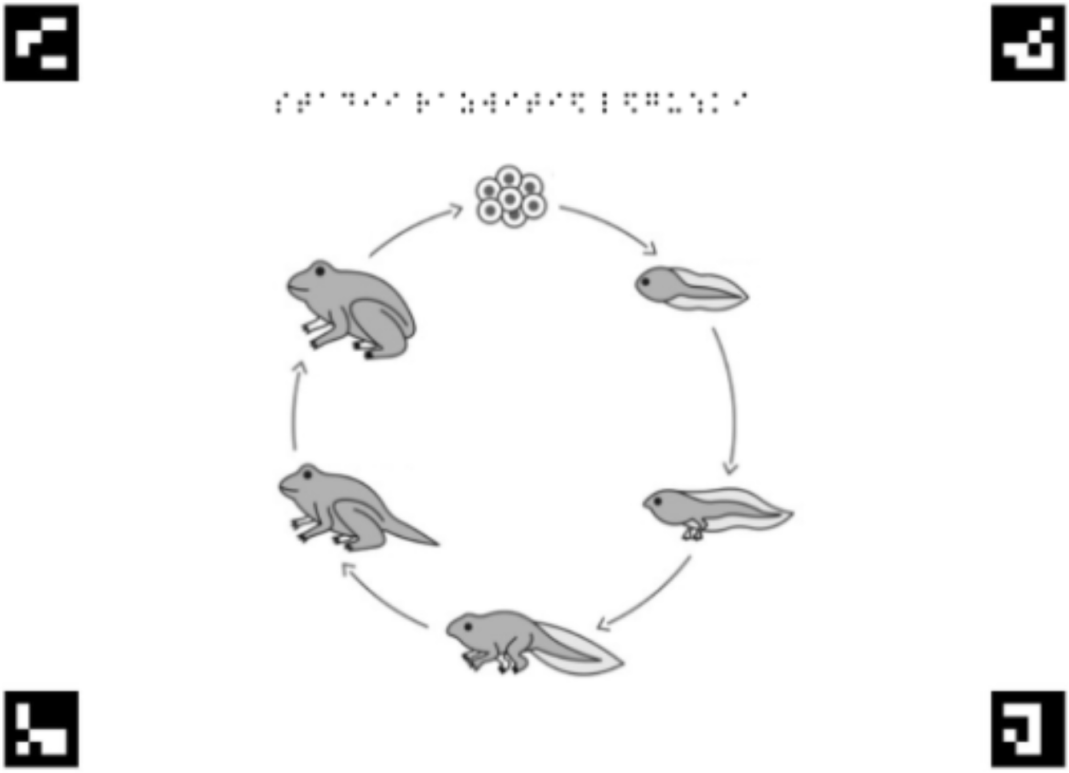


Figure 5.3: Example of TG used in the app exploration mode. No legends.

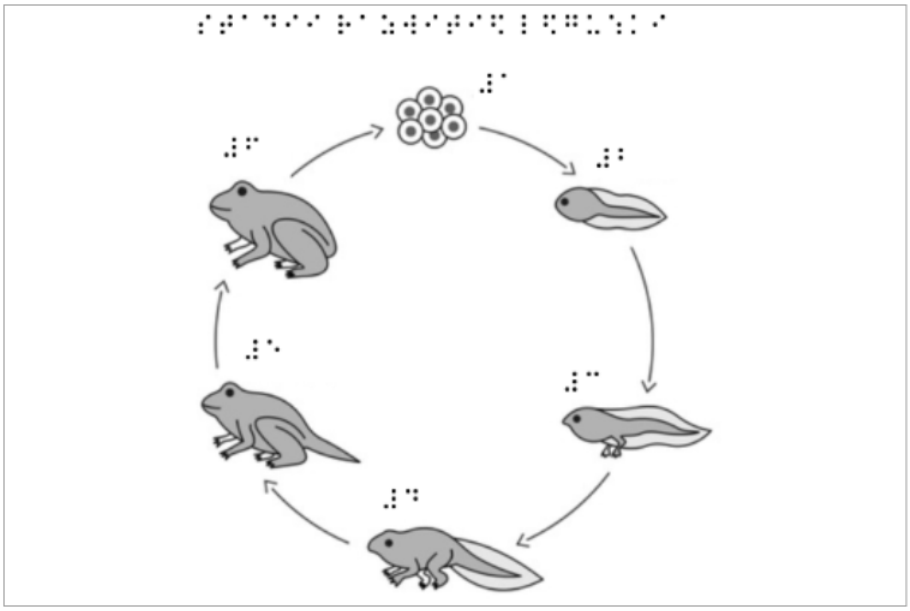


Figure 5.4: Example of TG used in the Braille and SR modes. With Braille legends

Figure 5.5: Participants exploring TG using the TAURIS app
Note: This photograph was taken for demonstration purposes only and was not taken whilst the participant was conducting an actual experiment

After the participants had familiarised themselves with the summary, the researcher read each question out loud and recorded the answers and response times. There were three general questions and one memory question for each object. The participants were free to continue exploring the graphic while answering the first three questions. The final memory question was used to assess how different exploration modes affect the participant's ability to remember the information. Since it was a test of memory, participants were not allowed to use the TG, Braille text information, or the App whilst answering this question. Participants were not told whether their answers were correct, as this is what happens in, for instance, an examination.

The following example shows the ordering of the questions from one session.

1. Object 1 (App)
2. Object 2 (SR/Braille)
3. Map 1 (App)
4. Map 2 (SR/Braille)
5. Graph 1 (App)
6. Graph 2 (SR/Braille)

The order of the questions in each task was the same for all participants. The order of the exploration modes always started with the app mode and was followed by the screen reader or the Braille mode. This was done because the order of the tasks in which the experiments were conducted could influence the performance, attitudes, and perceptions of the participants. For example, if a particularly difficult or time-consuming task was performed early in the study, it could tire the participants and affect their performance on subsequent tasks. On the other hand, if an easier task was conducted first, it could create a positive attitude that carried over to the rest of the study. This approach was used to ensure that all participants completed the tasks under the same conditions. The general design of the experiments is illustrated in Figure 5.6.

Memory Questions

The tasks and memory questions designed in this study serve to assess how effectively the proposed system supports visually impaired users in processing and retaining spatial and semantic information. An illustrative example includes a task where participants are required to recall and reconstruct the layout of a tactile map of Australia. This task not only demands an understanding of spatial relationships between states but also incorporates key geographic features, aligning with Petridou (2014) emphasis on interactive environments to sustain engagement and promote active learning. By fostering exploration, these tasks are intended to enhance cognitive engagement.

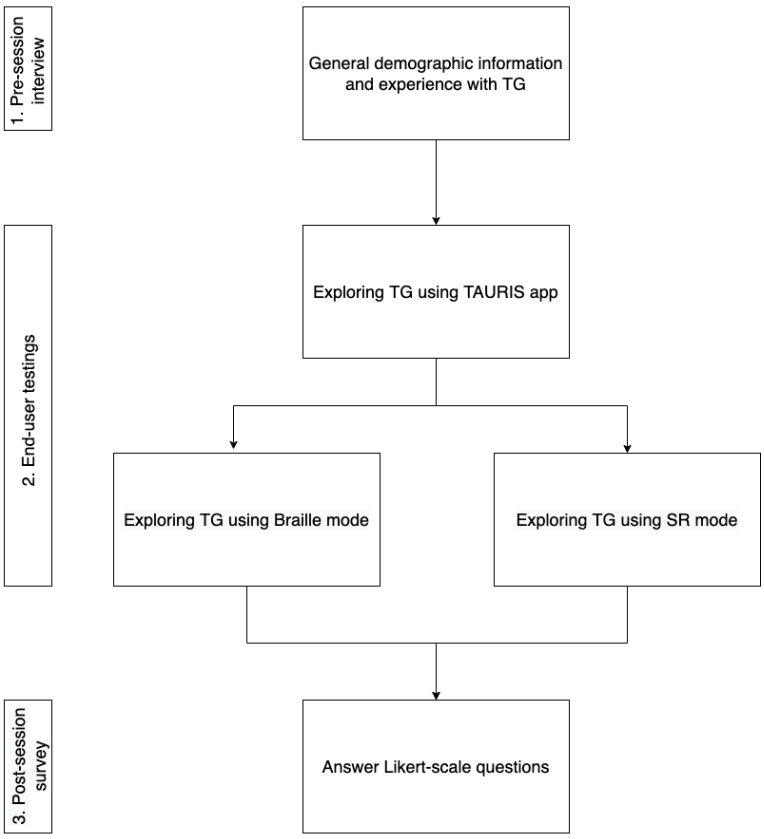


Figure 5.6: Experiments design

Memory questions were further designed to evaluate the integration of tactile and auditory information encountered during exploratory tasks. For instance, in the frog lifecycle task, participants were tasked with identifying developmental stages and reconstructing their sequence. This approach tests both retention and synthesis of multimodal inputs and is inspired by the methodology of Melfi et al. (2020), who demonstrated the efficacy of recall-based evaluation in assessing tactile graphics for educational purposes.

The design of these tasks is consistent with the broader objective of advancing accessible learning technologies for visually impaired individuals. By drawing from established frameworks in interactive environments and cognitive assessment, the study integrates tactile and auditory modalities to address the unique challenges faced by this demographic. This aligns with the aims of the TAURIS system, which leverages audio-tactile interactions to provide more intuitive and effective learning experiences. The descriptions of the tasks and the list of all questions can be found in Appendix B.

5.3.5 Phase 3 – End-user feedback (Qualitative)

The qualitative analysis was used to answer the first research question *"What are visually impaired individuals'* ***perceptions and attitudes*** *toward the use of smartphone app in the context of*

exploring Tactile Graphics (TG)?". A detailed understanding of user experience was gained by conducting structured interviews with open-ended questions. In addition, Likert scale questions were used to measure participants' attitudes and opinions.

Likert-scale questions

Following the experiments, the participants were asked to answer six Likert-scale questions (Likert, 1932). The answers ranged from 1 to 5, with 5- strongly agreeing and 1- strongly disagreeing. The corresponding statements were as follows:

1. The phone app helped me to understand concepts better than Braille or text descriptions alone
2. I found it easy and intuitive to use the app
3. I found it easy to properly aim the camera
4. I am satisfied with the app response time
5. I liked the detailed description app feature
6. I would be interested in using this app to get better quality information during my classes on a daily basis

Semi-structured interviews

Interviews are extremely helpful during product development. (Lazar, Feng, and Hochheiser, 2017, p. 195). As they allow a deeper insight into users' opinions about the product interface, information flow and other important features. There are three types of interviews:

- **Structured** interviews follow a predetermined set of questions and are designed to be more formal and standardised. The interviewer asks the same set of questions to all participants, and the responses are recorded verbatim.
- **Semi-structured** interviews also follow a set of predetermined questions, but the interviewer has the flexibility to ask follow-up questions or to deviate from the predetermined questions if necessary.
- **Unstructured** interviews do not follow a predetermined set of questions. The interviewer has the flexibility to ask any questions they see fit and to follow the conversation wherever it leads.

The semi-structured interview method was selected because it offers a balance between the structure of a structured and the flexibility of an unstructured interview. Also, it allows the interviewer

to probe more deeply into the participants' responses and to gather additional information. The addressed questions are listed below:

1. Please share any thoughts on your experience using the app.
2. Name at least three things you like about using the app.
3. Name at least three things you did not like about using the app.
4. Do you have any specific suggestions on which features might be added?
5. Do you feel like you can explain the concepts from the tactile graphic better now that you've used the app?

Semi-structured approach gave me room to ask interviewees to elaborate on some points. For instance, one of the participants said that he found the app very interactive. I asked him to comment on this and give an example. He replied that he particularly enjoyed exploring the maps, as it was easy to understand and remember the relative positions of the objects on a map. The answers to all questions were audio-recorded for further analysis.

5.3.6 Phase 4 – Interviews with teachers (Qualitative)

Phase 4 of the study aimed to gather valuable insights and feedback from teachers of visually impaired students regarding the potential integration and impact of the TAURIS system within educational settings. Five teachers who possessed a diverse range of expertise across different subject areas and grade levels were recruited.

Semi-structured interviews were conducted with each teacher, exploring their experiences and perspectives on utilising tactile graphics and assistive technologies in their classrooms. The interviews commenced by exploring the teachers' current practices and challenges associated with incorporating tactile graphics into their lessons. Specific examples were encouraged, such as difficulties encountered when conveying complex scientific concepts like optics in physics or intricate anatomical structures in biology. Further exploration focused on the potential benefits and applications of the TAURIS app within their teaching practices. Teachers were asked to reflect on how the app's features, such as real-time audio descriptions and interactive exploration, could enhance student engagement, comprehension, and independent learning. Additionally, feedback was sought on the TAURIS web tool, specifically its practicality and effectiveness in creating and annotating tactile images for classroom use. The interview concluded by inviting teachers to share any suggestions or recommendations they might have for further improving the TAURIS system to better cater to the diverse needs of visually impaired students and educators. The addressed questions are listed below:

1. Can you share your experiences with using tactile graphics in your classroom.

2. What are the challenges you face in using tactile graphics with visually impaired students? (Give particular examples, i.e. optics concept from physics)

3. How do you think the TAURIS app could benefit your teaching and your students' learning experience?

4. What are your thoughts on the TAURIS web tool for creating and annotating tactile images?

5. Do you have any suggestions for improving the TAURIS app and web tool?

5.4 Analysis

The quantitative and qualitative data analysis in my research was based on the procedures recommended by Creswell and Clark (2017, p. 210-212). According to the authors, there are six steps to follow: data preparation, data exploration, data analysis, analysis representation, results interpretation and results validation. The actions described below highlight the key elements of both quantitative and qualitative analyses used in this research.

1. Data preparation

Preparing data for quantitative analysis involved transferring the numerical results of the experiments to Excel charts. I separated the response times and the percentages of correct answers into different files. Within the files, values were stored in different charts depending on the type of mode, level of vision loss, and age of the participant.

Data preparation in a quantitative strand took longer because the interviews were carried out in local languages: either Kazakh or Russian. After the interview, recordings were transcribed verbatim by myself. I translated these transcriptions into English.

2. Data exploration

In order to explore quantitative data, descriptive statistics were used, with all statistical analyses being performed in Python. One of the main challenges was to identify the right statistical method for the data analysis. After performing a series of Shapiro-Wilk tests (Shapiro and Wilk, 1965), it was verified that most of the data were not normally distributed.

Qualitative data was explored by thoroughly reading the transcriptions and taking notes in the margins. This was necessary to develop the initial codes for further analysis.

3. Data analysis

Since quantitative data was not normally distributed, non-parametric statistics were used. In particular, the Wilcoxon signed rank test was used for the within-subjects design to test the

significance of the differences between the means of two groups. On the other hand, the Mann-Whitney U test was used for the between-subjects designs to test the effect of age and vision loss. SciPy STATS module was used to find the significance scores (Virtanen et al., 2020).

The qualitative analysis process began with deductive coding. Deductive coding is a top-down approach, where codes are derived from a pre-existing theoretical framework or research question. The Delve[5] online tool was used for data indexing and categorising. This helped me develop theories and concepts and then group them into themes and sub-themes in a more efficient way.

4. Data representation

Data representation for quantitative analysis involved summarising the results of statistical tests in statements and tables.

For qualitative data representation, I provided quotes from the interview transcriptions. The cited quotes were organised into themes and sub-themes.

5. Results interpretation

This step involved analysing the quantitative results to directly address the research questions and assess the validity of the corresponding hypotheses. It also involved investigating the sources of uncertainties and biases in the results.

Qualitative data interpretation also involved comparing results with research questions. In addition, I have performed a personal assessment of the findings and related them to the existing literature. Overall, data interpretation in both quantitative and qualitative research involves making sense of the data collected and using it to draw conclusions and develop insights about the research topic.

6. Data and results validation

Data validation is one of the essential components of good research practice (Denscombe, 2017). It helps to ensure the reliability and validity of the data and to improve the trustworthiness of the research findings. Triangulation is one of the techniques used in mixed-methods research to strengthen the credibility of the data and results. This technique involves "the use of multiple data sources with similar foci to obtain diverse views about a topic or a purpose of validation" (Kimchi, Polivka, and Stevenson, 1991, p. 365). The forthcoming chapter will present side-by-side comparisons of the outcomes obtained from the quantitative and qualitative research strands. Additionally, the principles outlined in the "Good Research Practice Guide" (Denscombe, 2017) were followed throughout the research process.

[5]https://delvetool.com/

5.5 Conclusion

In this chapter, the problem definition, methodological approach, and data collection procedures were presented. To evaluate the performance of the TAURIS App, an initial phase of data collection involved 12 experimental sessions (including one pilot session) with visually impaired individuals. Participants were first asked about their general demographic information (age, gender, level of sight loss, etc.) and their prior experience with TG. They were then given time to familiarise themselves with the app before proceeding to explore tactile graphics using two different methods: the TAURIS app and either Braille texts or a screen reader. The number of correct responses and the time spent on each task were recorded for analysis. To enhance the statistical power of the findings and gather a more comprehensive dataset, an additional 9 secondary school students were recruited and interviewed during the school term, resulting in a total sample size of 20 participants. The next stage of the study involved gathering feedback from both the initial participant group and the additional students through Likert-scale questionnaires and open-ended interviews, focusing on their experiences with the app. Furthermore, Phase 4 of the study involved engaging with teachers of visually impaired students to gather their insights and perspectives on the potential integration and impact of the TAURIS system within educational settings.

A convergent design mixed-method data analysis approach was employed to interpret the results, allowing for a deeper and complementary understanding of the topic. The quantitative and qualitative data from the initial phases and the teacher feedback were analysed separately to ensure a comprehensive examination of the findings. The subsequent chapter will present and discuss these findings in detail.

Chapter 6

Results and Discussion

6.1 Introduction

In this chapter[1], the key findings of the research are presented and discussed. The participant demographics and background information are provided in Section 6.2. In Section 6.3, the outcomes of the interviews, in which participants shared their experience with Tactile Graphics (TG) and mobile devices, are reported. The quantitative results of the app testing with end-users are presented in Section 6.4. In Section 6.5, the end-user session results, including Likert-scale question responses and open-ended interview findings, are presented. Additionally, the results of merging the two data strands are described in this section. Section 6.6 presents the findings obtained from teachers' interviews. Following the presentation of the results, the chapter moves to a comprehensive discussion in Section 6.7. Finally, the chapter is summarised and concluded in Section 6.8.

6.2 Participants

The initial phase of the study involved 11 participants (excluding the pilot study participant) who were interviewed during the summer holidays. To enhance the statistical power of the findings and gather a more comprehensive dataset, an additional 9 secondary school students were recruited and interviewed during the school term. This resulted in a total sample size of 20 participants (11 males and 9 females) with ages ranging from 17 to 65 years (average = 27.62 years, SD = 12.54). The decision to include participants over the age of 16[2] was made in consideration of the practical constraints imposed by the lockdown measures in place during the initial phase of data collection. Furthermore, the inclusion of school staff and alumni alongside students

[1]Some of the work in this chapter has appeared in Zeinullin and Hersh (2022). Maralbek Zeinullin is the first author and main contributor to this paper.

[2]The threshold for the young participants recruitment in the University of Glasgow is 16 years and the inclusion of younger individuals necessitates the fulfilment of additional legal obligations.

allowed for a broader representation of experiences within the visually impaired community.

Nine participants were secondary school students, and three were university students. Six participants were employed at the school library (Shymkent regional boarding school for visually impaired children), one was a teacher at the same school and one was working as a masseur. Information about the schools they attended was collected as well. Nine participants currently attended the school, six had attended it in the past, four participants previously attended mainstream school and one participant attended both schools.

Nine participants identified themselves as blind and eleven had low vision. Five participants had been visually impaired from birth, six lost their vision in childhood, and the remaining after the age of eleven. Sixteen participants were Braille literate and four said they understood some Braille but were not fluent. Table 6.1 summarises the participant information.

ID	Gender	Age	Occupation	Vision	Braille literate	School attended
1*	Male	27	Library worker	Visually impaired	Yes	School for the blind
2	Female	42	Library worker	Blind	Yes	Mainstream school
3	Male	36	Library worker	Visually impaired	No	Mainstream school
4	Female	28	University student	Visually impaired	No	Both
5	Female	35	Library worker	Blind	Yes	Mainstream school
6	Female	45	Library worker	Visually impaired	Yes	School for the blind
7	Male	28	School worker	Blind	Yes	School for the blind
8	Male	35	Masseur	Blind	No	School for the blind
9	Male	18	School student	Visually impaired	Yes	School for the blind
10	Male	18	School student	Visually impaired	Yes	School for the blind
11	Male	37	Library worker	Visually impaired	Yes	Mainstream school
12	Female	65	Library worker	Blind	Yes	School for the blind
13	Male	21	University student	Blind	Yes	School for the blind
14	Male	22	University student	Blind	Yes	School for the blind
15	Male	17	School student	Visually impaired	Yes	School for the blind
16	Male	18	School student	Visually impaired	No	School for the blind
17	Female	17	School student	Blind	Yes	School for the blind
18	Female	19	School student	Visually impaired	No	School for the blind
19	Male	18	School student	Visually impaired	Yes	School for the blind
20	Male	17	School student	Blind	Yes	School for the blind
21	Female	17	School student	Visually impaired	Yes	School for the blind

Table 6.1: Participants information
*Note: Pilot study participant was not included in the analysis

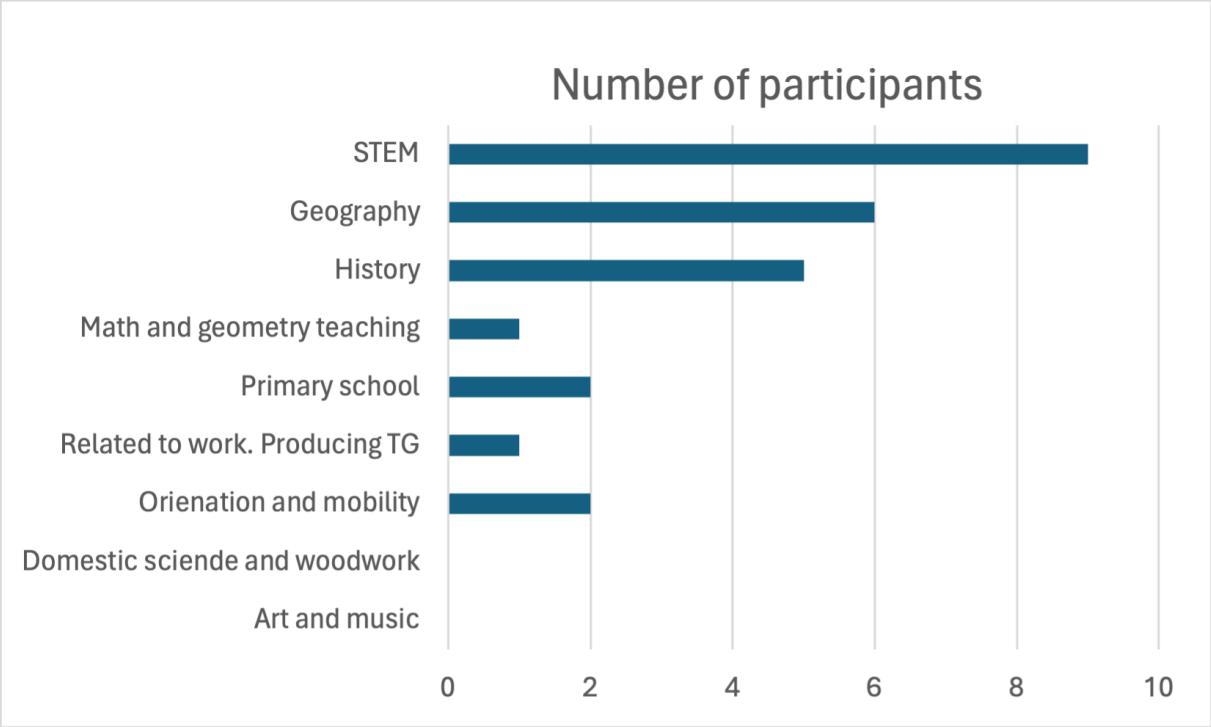


Figure 6.1: In which subjects TG were used

6.3 Phase 1 - Interview

6.3.1 Experience with the tactile graphics

Regarding their experience with the TG, twelve participants said that they used such graphics at school and eight have not. TG was used primarily in science technology engineering and mathematics (STEM), geography and history subjects and were labelled with Braille text. More detailed information is presented in Figure 6.1. All of those who had an experience with the TG reported that it was easier to understand the subject when such graphics were used in class. For example, TG helped them learn the computer keyboard. In particular, they used the tactile version of the keyboard to learn the spatial location of the keys. Also, during chemistry classes, a tactile version of the periodic table was used to present the material in an easier-to-understand format. One of the disadvantages was that it required a teacher to show and navigate through the TG individually to each student first.

6.3.2 Experience with mobile devices and applications

During the study, participants were asked about their usage of mobile devices. Thirteen participants had Android and seven had iOS smartphones. Nine participants reported using a phone camera on a daily basis, while five participants stated they used it once or twice per week. Additionally, two participants reported using a phone camera once a month, and the remaining four

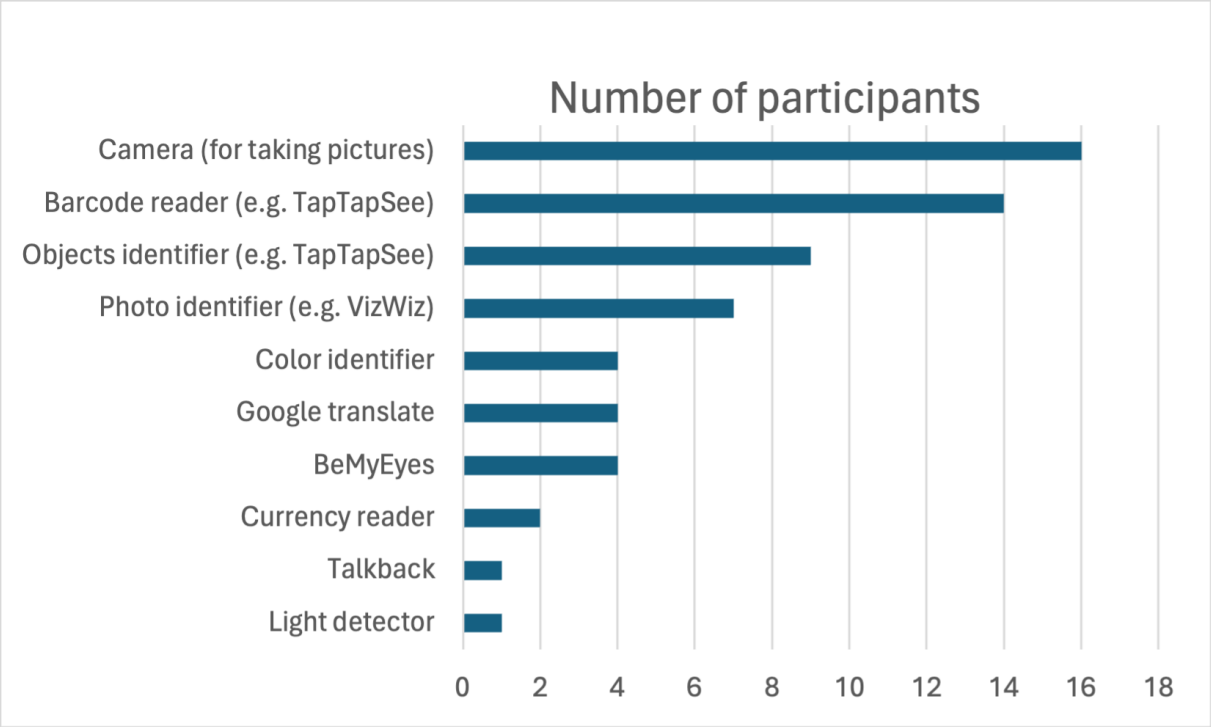


Figure 6.2: Apps used by participants

participants stated they never used it. Most of the participants (16/20) used a camera to take photos. Apps which required a camera, and which were used by the participants were: currency and barcode readers, colour and photo identifiers and light detectors. Other popular apps were BeMyEyes, TapTapSee and Google Translate (Figure 6.2). The primary challenge identified in relation to these apps was proper camera aiming. All participants who reported using a phone camera noted difficulties with this task. The proper utilisation of a phone camera by individuals who are blind has been a longstanding issue and various researchers have attempted to address it. This problem is discussed in greater detail in Section 2.2.3 of this thesis.

6.4 Phase 2 - Device testing

6.4.1 Comparison of App and Braille text

Mode	App	Braille
Average time (sec)	43.49	72.05
Average accuracy (%)	93.67	86.10
Memory accuracy (%)	88.00	81.86

Table 6.2: App vs Braille text

As mentioned above, sixteen participants said that they were Braille literate. Ten of them were randomly selected to utilise Braille texts during the experiments. Participants explored the TG and were asked questions about it. On average, participants were able to provide answers quicker and with greater accuracy while exploring the TG using the app compared to Braille (Table 6.2).

The results of the non-parametric Wilcoxon signed-rank test indicated that the TG exploration mode had a statistically significant impact on both the **time spent** answering the questions (W = 6.0, p = 0.0002, N = 10) and the **average accuracy** achieved (W = 5.0, p = 0.023, N = 10). However, the difference in **memory accuracy** between the two groups remained statistically insignificant (W = 35.0, p = 0.361, N = 10).

I also analysed the impact of participants' ages on their performance. Their average age was about 27 years. They were therefore divided into two groups: over 27 (n = 9, mean = 35) and under 27 (n =11, mean = 18). From Table 6.3 it can be seen that the under 27 age group was answering faster but showed worse performance in remembering the information in both exploration modes. In the app exploration mode, the under 27 age group demonstrated a slightly higher percentage of correct answers. Conversely, in the Braille mode, the over 27 age group exhibited a slight advantage in accuracy. A series of non-parametric Mann-Whitney U tests were conducted to examine the impact of age on time, accuracy, and memory accuracy. The results indicated that the observed differences were not statistically significant. Specifically:

- For **time**, the Mann-Whitney U test yielded $U = 45.0$, $p = 0.260$.
- For **accuracy**, the test produced $U = 50.5$, $p = 0.937$.
- For **memory accuracy**, the test reported $U = 44.0$, $p = 0.262$.

These findings suggest that age did not have a statistically significant effect on any of the measured outcomes.

Nine participants indicated that they were blind, either totally blind or blind and have light perception. Eleven participants stated that they were able to distinguish shapes and read very large print texts, classified as partially sighted. A comparative analysis of the two groups revealed following findings regarding the impact of visual impairment on performance. In the app exploration mode, the time spent answering questions was nearly identical for both blind and visually impaired participants (Table 6.3). However, a contrasting trend emerged in the Braille mode, where visually impaired individuals demonstrated faster response times. Accuracy remained consistent across both groups and exploration modes. Interestingly, the visually impaired group exhibited slightly higher memory accuracy in the Braille mode. Despite these observed trends, the Mann-Whitney U tests indicated no statistically significant differences between the groups for any of the measured variables. Specifically:

- For **time**, $U = 48.5$, $p = 0.88$.

- For **accuracy**, $U = 46.0$, $p = 0.70$.
- For **memory accuracy**, $U = 50.0$, $p = 0.99$.

Mode	**App**		**Braille**	
Age	**Under 27**	**Over 27**	**Under 27**	**Over 27**
Time spent (sec)	38.80	50.52	68.89	72.29
Average accuracy (%)	95.00	91.67	84.82	88.00
Memory accuracy (%)	85.56	91.67	79.56	85.33
Vision	**VI**	**Blind**	**VI**	**Blind**
Time spent (sec)	43.82	43.16	62.13	81.97
Average accuracy (%)	94.98	92.35	86.08	86.10
Memory accuracy (%)	87.20	88.80	79.33	84.40

Table 6.3: Comparison of different age and vision impairment groups

6.4.2 Comparison of App and Screen Reader

Mode	**App**	**SR**
Average time (sec)	50.75	62.54
Average accuracy (%)	96.35	82.12
Memory accuracy (%)	92.47	75.53

Table 6.4: App vs Screen Reader

Table 6.4 illustrates that participants exhibited faster response times and achieved higher accuracy, including memory accuracy, during the app exploration mode compared to the screen reader mode. The Wilcoxon signed-rank tests confirmed the statistical significance of these differences for all three measured variables: **time spent** (W = 2.0, p = 0.02, N = 10), **average accuracy** (W = 1.0, p = 0.004, N = 10), and **memory accuracy** (W = 1.0, p = 0.03, N = 10).

As in the previous section, after running the Mann-Whitney U tests, it was found that differences between age and vision loss groups were not statistically significant. Specifically, for the **impact of age**:

- **Time**: $U = 43.0$, $p = 0.367$.
- **Accuracy**: $U = 47.5$, $p = 0.797$.

Mode	**App**		**SR**	
Age	**Under 27**	**Over 27**	**Under 27**	**Over 27**
Time spent (sec)	52.15	38.17	65.80	33.25
Average accuracy (%)	96.42	95.83	85.28	97.17
Memory accuracy (%)	93.48	83.33	74.07	88.67
Vision	**VI**	**Blind**	**VI**	**blind**
Time spent (sec)	56.18	45.94	62.28	60.02
Average accuracy (%)	96.27	96.48	83.89	79.48
Memory accuracy (%)	91.16	93.75	70.28	83.42

Table 6.5: Comparison of different age and vision impairment groups

- **Memory Accuracy**: $U = 45.0$, $p = 0.544$.

For the **impact of vision loss**:

- **Time**: $U = 44.0$, $p = 0.380$.
- **Accuracy**: $U = 48.0$, $p = 0.773$.
- **Memory Accuracy**: $U = 42.5$, $p = 0.414$.

6.5 Phase 3 - End-user feedback

The outcomes of the qualitative analysis are presented in this section. The audio recordings of the open-ended interviews were transcribed. The obtained data was analysed with deductive coding approach. Since the interviews were conducted in Russian and Kazakh languages, they were translated before the analysis.

6.5.1 Likert-scale questions

An evaluation utilising a Likert scale was conducted to investigate respondents' attitudes toward the app. According to the results presented in Table 6.6, a majority of the users were satisfied with the application. The overwhelming majority, 18 of 20 (90%) either agreed or strongly agreed that they would be interested in using this app on a daily basis. Also, 12 out of 20 (60%) agreed that it was easy to aim the camera while using the app.

Statement	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
The app helped me to understand concepts better than Braille or text descriptions alone	13	2	2	3	0
I found it easy and intuitive to use the app compared to Braille or text descriptions alone	5	12	1	2	0
I found it easy to properly aim the camera	5	7	2	6	0
I am satisfied with the app response time	5	10	1	4	0
I liked the detailed description app feature	13	4	1	1	1
I would be interested in using this app to get better quality information during my classes on a daily basis	15	3	1	0	1

Table 6.6: Results of the Likert scale question

6.5.2 Semi-structured interviews

Participants commented on app effectiveness in response to the open-ended interview questions. I have employed a deductive coding approach in alignment with the research questions. Through this analysis, four main themes emerged: speed and ease of use, ability to remember, users perceptions of the app and comprehension of the concept. Finally, users' suggestions were recorded to improve the existing system.

Speed and ease of use

P2 stated:

"I liked the speed of working with the app and the QR code image identification feature. This makes the whole process very convenient and fast."

This view was echoed by P7 who said:

"I really liked the app because it makes it very convenient to use tactile images. The user doesn't have to spend time and read or look for the descriptions. The app tells everything in real-time."

P5 supported this as well:

"For instance, if we compare this to Braille text descriptions, it was much faster to explore the TG by using the app."

There were also some negative comments about the application. P8 said:

"Sometimes the app failed to detect the finger. So, I had to wait for the audio descriptions. I noticed that this occurred when the finger was located on the border between two objects so the app kept jumping between two descriptions. However this happened very rarely."

The qualitative data demonstrates the app's potential to significantly enhance the TG exploration experience. Users consistently highlighted its speed, convenience, and real-time information delivery. The ability to quickly access information through QR code and tactile image recognition proved particularly valuable, streamlining the process compared to traditional methods.

While the occasional finger detection issue raised by P8 recommends further investigation and refinement, the overall feedback strongly suggests that the app effectively addresses the needs of VIP seeking a faster and more intuitive way to engage with TG.

Ability to remember

The comments below illustrate how the app affects the users' ability to remember information. For instance, P11 stated:

"I think the app helps the user to develop his/her spatial thinking. Thus, it is easier to draw the connections between objects and remember the information provided."

This statement echoed by P10: *"It was also very convenient to explore maps with the app. It helped me to construct a 2D image in my mind. I think this will be very helpful for the learners in schools."*

The qualitative data indicates that the app's interactive nature and ability to provide simultaneous tactile and auditory feedback contribute to improved spatial understanding and memory retention. Users reported feeling more confident in their ability to recall information after exploring TG through the app, particularly when it came to understanding relationships between objects on a map.

Users' perceptions

The following participant comments provide insights into their perceptions of the app and thus contribute to addressing Research Question 1. P2 commented:

"The first impression is very positive. Despite the fact that this is just a beta version of the app, it works smoothly. I want this app to be widely used as soon as possible."

This echoed in P4:

"I really liked the app. I think it will definitely benefit the visually impaired community."

P5 said:

"App is very good and works fast. I like that this system was built on a mobile platform, it is very convenient. In addition, I liked the app's interactivity. It makes the learning process more exciting"

Figure 6.3: Interviews transcriptions word cloud

P6 shared his thoughts as well:

"The application is very convenient. Many blind people cannot read Braille. Especially the ones who became blind in adulthood. Since it is much more difficult to learn Braille at that age, the only way for them to acquire information is through audio format. Also, not everyone among VIP can properly read the tactile images. Therefore, real-time audio descriptions provided by the app are a very handy solution."

Some participants also saw a need for changes, P8 said:

"The app works well but there is certainly room for improvement."

The interview results reveal a strong sense of optimism and enthusiasm surrounding the app's potential. Users mentioned its accessibility, ease of use, and ability to provide valuable information in a format that is readily understandable. The app's mobile platform and interactive features were particularly well-received, suggesting a strong alignment with user needs and preferences.

Concept comprehension

P14 said:

I think so. The frog lifecycle was tough. I knew there were several stages, but I couldn't really explain how a tadpole changes into a frog. The app helped me understand that. Now, when I touch the tadpole, I hear the app describing its features and then how it develops legs and loses its tail. I can explain that process much more clearly now.

P17 commented: *I can now explain the map with more confidence. Before, I'd just say, 'Australia is a big island with lots of states.' Now, I can talk about the different states and know the relative location of Tasmania and even its capital. Before I had to go back and forth and check the names. So, I would definitely forget the capital by the time I got back to the tactile map.*

P19 had mixed feelings about the app" *"The app was great for getting a general understand-*

ing of the space shuttle. And this was what I knew already. I could feel the wings and engines and hear what they were, and it helped me picture the whole thing. But I think it will be more challenging to incorporate a smaller details, like the parts of engine. The exploration area on the tactile graphic is too limited to effectively represent those smaller parts."

The qualitative results suggest that the app's multi-sensory approach can significantly enhance understanding, particularly for complex concepts with multiple stages or intricate details. Users reported feeling more confident in their ability to explain and recall information after interacting with the app. However, the feedback also highlights the importance of considering the scale and complexity of the TG.

Users' suggestions

P2 had several suggestions:

I think it will be better if the app will be capable of detecting a new QR code automatically. So, the user does not have to close and open the app again in order to start working with a new tactile image. Also, I propose to use a universal frame with markers in the corners and where the tactile image can be placed. Also, it would be great to develop IOS and Windows versions of the app.

P7 proposed:

Finger-pointing feature would be helpful. For instance, I'd like the app to say move left if the object is there. Also, I think it will be helpful to notify the user if the level of illumination in the room is too low.

P9 had comments on the audio output:

Add Kazakh language and switch to the Google voice synthesizer instead of the Samsung one.

P10 said:

Option to increase the speed of speech. Utilisation of the wider angle camera, so a bigger image can be explored. Adding the Kazakh language

Finally, P12 suggested:

I think the objects' properties, like colour and textures should be added to the object descriptions. Also, it would be helpful to fill the objects with different lines and dashes

The user feedback provides a valuable roadmap for future development, highlighting areas where the app can be further improved to enhance its usability, accessibility, and overall user experience. The suggestions cover a range of functionalities, from technical improvements like automatic QR code detection and platform expansion to content enhancements like object property descriptions and tactile differentiation.

Addressing these suggestions will not only improve the app's functionality but also demonstrate a commitment to user-centered design, ultimately leading to a more valuable tool for the VIP.

6.5.3 Merging Quantitative and Qualitative Data

The merging of qualitative and quantitative data is a widely used research method that provides an in-depth understanding of the research topic. Qualitative data is often used to provide a comprehensive understanding of a phenomenon by giving context and background information. Quantitative data, on the other hand, is used to identify trends and patterns in the information. Merging the two types of data allows to draw more meaningful conclusions.

The **joint display** is a useful approach that enables the merging of qualitative and quantitative data and thus provides a more comprehensive view of the study results (Creswell and Clark, 2017, p. 228). It is used to display the results of both quantitative and qualitative stands together in a single display, providing an integrated perspective of the research findings. In a tables presented below the quantitative data is presented in boxplots to provide a comprehensive view of the data distribution. Boxplots allow for analysis beyond central tendency, as they display measures of spread, skewness, and potential outliers. This provides a more informed understanding of the data compared to relying on a single summary statistic, such as the mean. Qualitative data is represented in the form of quotes from interviews. This method has the potential to reduce bias in the interpretation of the results and allows for a clearer comparison of the two different types of data. Legocki et al. (2015) and Beck, Eaton, and Gable (2016) used this technique to represent mixed data in their research as well. In my research, I have used this method to answer the first two research questions.

Research Question #1

What are visually impaired individuals' perceptions and attitudes toward the use of smartphone app in the context of exploring Tactile Graphics (TG)?

Figure 6.4 presents a joint analysis of quantitative Likert-scale data and qualitative insights from participant interviews, specifically addressing the question of visually impaired individuals' perceptions and attitudes toward using a smartphone app for TG exploration. The left panel illustrates the distribution of Likert-scale responses to key statements about the app, where a range of 1 to 5 indicates levels of agreement, from 'Strongly Disagree' to 'Strongly Agree'. The analysis shows a clear trend: a majority of participants expressed a positive attitude towards the app, viewing it as a superior method for TG exploration compared to Braille text and screen readers. Furthermore, a significant portion of users indicated that they were satisfied with the real-time nature of the system. This preference was supported by participants' enthusiasm towards using the app, with many expressing an interest to utilize it in their daily learning. The qualitative data, presented in the right panel through representative participant quotes, reinforces these findings. Users highlighted the app's perceived convenience and speed, noting that real-time audio feedback eliminated the need to spend time searching for the descriptions. They also emphasized the interactive nature of the app and their feeling that the app promotes a more convenient and less cumbersome approach to TG access. These consistent findings from both

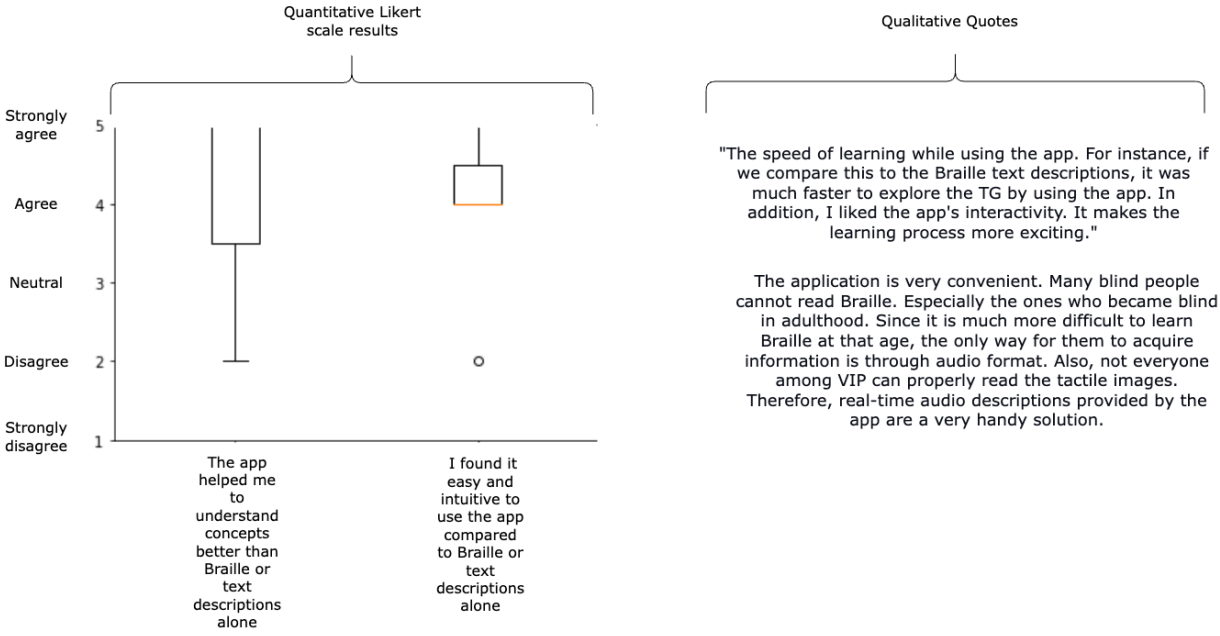


Figure 6.4: Joint Display of QUANT and QUAL data of what mode users prefer

quantitative and qualitative data confirm that, on the whole, visually impaired individuals perceive smartphone-based apps as having positive attributes in terms of usability, convenience and the promotion of independence when exploring TGs.

Research Question #2

To what extent does real-time speech output, integrated with tactile exploration, enhance the comprehension and retention of complex information conveyed through TG for visually impaired users?

The effectiveness of real-time speech output in enhancing comprehension and retention was evaluated by examining task completion time, accuracy, and memory recall. Figure 6.5 displays the distribution of task completion times using boxplots. These plots depict the lower quartile, median, upper quartile, and range of completion times. The results clearly show that participants using the TAURIS app, which integrates real-time audio descriptions, consistently completed tasks faster than when using Braille texts or screen readers ($p < 0.05$). This was supported by the participants' opinions that the real-time nature of the app enabled faster information acquisition, as illustrated by their quotes in the figure's right-hand panel.

The impact on comprehension was evaluated through accuracy scores, as shown in Figure 6.6. The boxplots reveal that users of the TAURIS app provided more accurate answers to the questions about the TG than those using traditional methods. These findings were statistically significant as well ($p < 0.05$). Qualitative comments further highlight that audio feedback that is directly linked to the exploration, which allows participants to navigate and gain information more effectively. Furthermore, as shown in Figure 6.7 participants who explored TGs using the

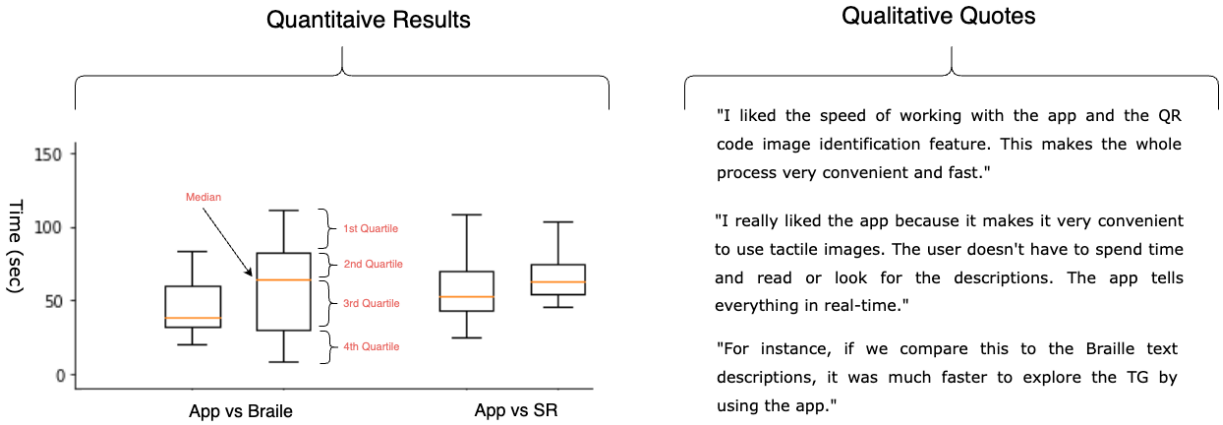


Figure 6.5: Joint Display of QUANT and QUAL data of time analysis

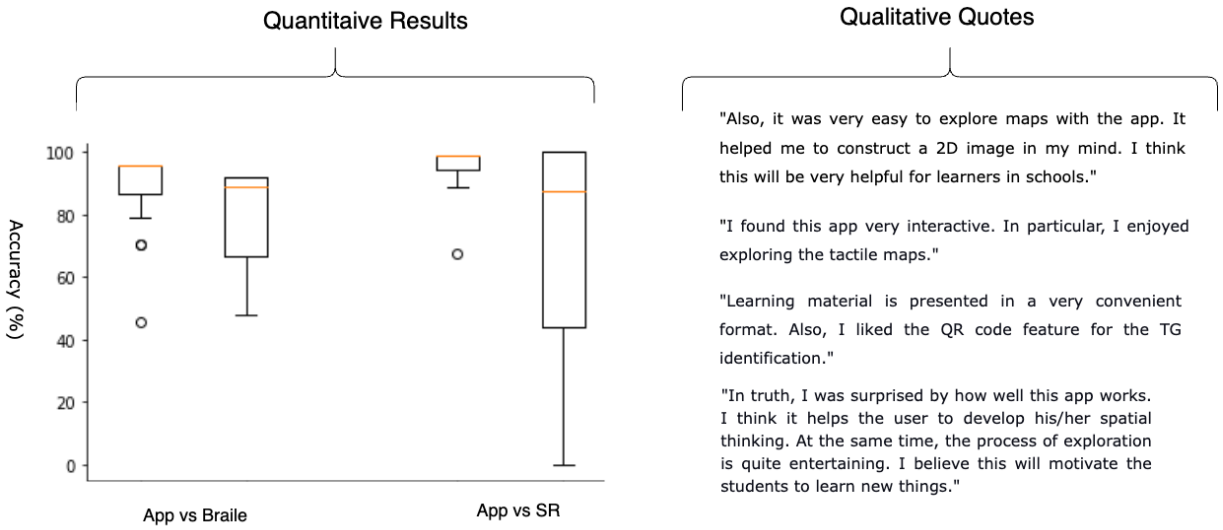


Figure 6.6: Joint Display of QUANT and QUAL data of accuracy analysis

TAURIS app responded with greater accuracy when asked to answer memory based questions compared to other approaches. However, the difference between the app and Braille text modes did not reach statistical significance (p=0.361). To support the observed quantitative results, associated participant quotes are added on the right side of Figure 6.7. They provide insight into the app's facilitation of users to form a '2D image' of the information and how the information is retained more effectively through combining audio and tactile exploration modes. Overall, these results suggest that the integration of real-time speech output with tactile exploration not only leads to faster learning but also improves comprehension and retention of complex information for visually impaired individuals.

Research Question #3

What methods can be employed to improve camera aiming in smartphone-based assistive technology applications designed for exploring TG?

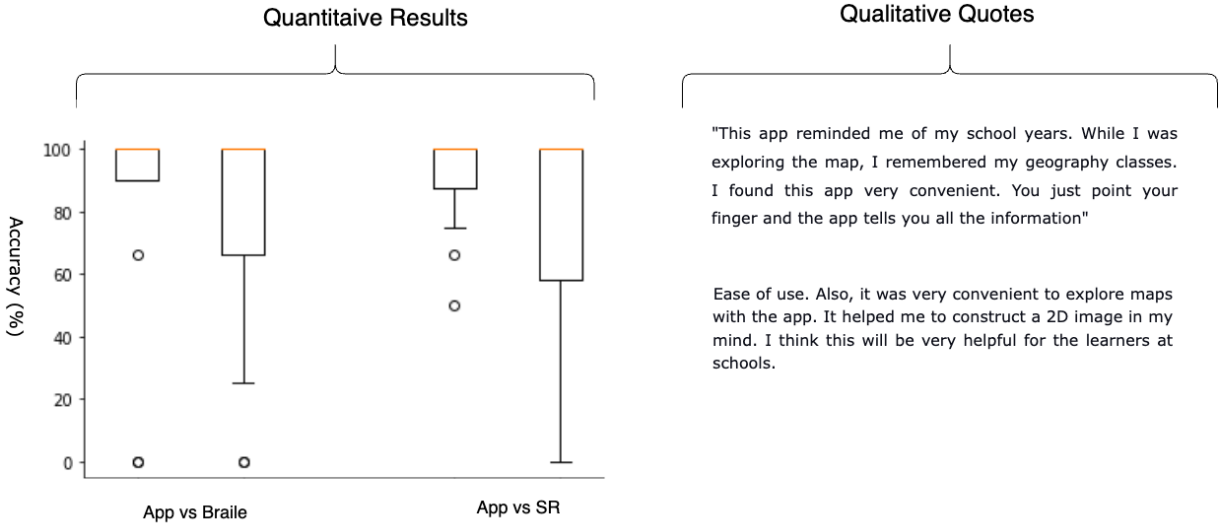


Figure 6.7: Joint Display of QUANT and QUAL data of memory accuracy analysis

The implementation of ARUCO markers combined with vibration feedback in the TAURIS app yielded the following results regarding camera aiming for visually impaired users. Prior to conducting the experiments, the participants were instructed to mount the mobile phone on a phone holder and align its camera accurately with the tactile graphic (TG) placed in front of them. The results of the Likert-scale questions indicated that 12 out of 20 participants found it easy to align the camera using this setup. Furthermore, the study achieved a 100% success rate in proper aiming. All participants in the user study successfully achieved proper camera alignment using the ARUCO markers integrated with the app's real-time vibration feedback feature. Subjective user feedback from interviews further validated these results, with participants indicating that the markers and vibration feedback provided clear cues for camera aiming and that they were confident in their ability to align the TG properly with the phone camera. These results show the effectiveness of the proposed system in ensuring accurate and efficient camera aiming for the participants.

6.5.4 Summary

The findings of this study offer key insights into the use of mobile assistive technology for TG exploration. Specifically, a combined analysis of Likert-scale and qualitative interview data revealed a positive user perception of the TAURIS app, with participants expressing a preference for its convenience, speed, and interactivity compared to traditional Braille or screen readers. Furthermore, incorporating real-time audio descriptions with tactile exploration resulted in statistically significant improvements ($p<0.05$) in response time, accuracy, and memory retention compared to screen reader mode, and significantly faster response times and higher accuracy in compared to Braille text mode ($p<0.05$) while showing similar memory retention. Qualitative feedback showed users found real-time audio particularly helpful for building spatial under-

standing and articulating complex concepts with confidence. Finally, a camera-aiming system, leveraging ARUCO markers and vibration feedback, proved fully effective, with all users in the study achieving correct alignment using this method.

6.6 Phase 4 - Interviews with teachers

This section presents the results obtained from interviews with teachers of visually impaired students. I believe these finding will provide valuable insight on the potential integration and impact of the TAURIS system within educational settings. Five teachers, each with a diverse range of expertise across different subject areas and grade levels, were recruited for this phase of the study. Information about teachers can be found in Table 6.7. All interviews were conducted in Russian, as this language was most comfortable for the participating teachers. Subsequent to the completion of the interviews, all recordings were transcribed verbatim and translated into English. Using a deductive coding approach several themes emerged from the analysis. These themes, representing the core insights gained from the teachers' interviews, are presented in the following sections.

The Value of Tactile Graphics in Specific Subjects

Chemistry teacher said:

"Tactile graphics are essential for teaching chemistry to visually impaired students. Chemistry involves understanding the structure and interactions of molecules, which are inherently spatial concepts. Tactile graphics allow students to explore these concepts through touch, which can help them to develop a deeper understanding of the material."

Physics teacher commented:

"When teaching about electric circuits, I use tactile diagrams of circuits with different components like batteries, resistors, and capacitors. Students can explore the layout of the circuit by touch and feel the raised lines representing the wires and components. This helps them to visualize the flow of electricity and understand how the different components interact."

Challenges to Tactile Graphic Integration

However, there were some challenges associated with the TG representation.

Physic teacher pointed out:

"Optics can be a challenging topic to teach to visually impaired students because many of the key concepts are inherently visual, such as reflection, refraction, and image formation"

Chemistry teacher added:

"It's difficult to show dynamic processes, like chemical reactions, using tactile graphics. Students need to be able to visualise the movement of atoms and molecules as bonds form and

break."

Also, all teachers agreed that creating tactile graphics is a time-consuming task. Biology teacher said:

"Creating and adapting tactile graphics can be time-consuming and require specialised skills. There are not always readily available resources for the specific topics I'm teaching, so I often have to create my own graphics."

The Potential of TAURIS

Math teacher said:

"I think the TAURIS app has the potential to be a very helpful tool for both teachers and students. The app makes it easier to create and access tactile graphics, and the real-time audio descriptions provide additional support for students who struggle with tactile perception."

Geography teacher commented:

"The app's two-handed exploration feature is fantastic. It's much more natural and intuitive for students to be able to use both hands to explore the graphics. It helps them to build a more complete mental image."

Suggestions for Improvement

Physics teacher suggested:

"It would be helpful to have a library of pre-made tactile graphics for common physics concepts."

Informatics teacher proposed:

"More customisation options for the audio descriptions would be beneficial, such as the ability to adjust the language, speed and voice of the speech output. And maybe even the option to have a voice that reads more expressively."

ID	Gender	Age	Subject	Years of experience
1	Female	56	Informatics	23
2	Female	35	Physics	11
3	Female	47	Math and Geometry	20
4	Female	64	Biology	42
5	Female	33	Chemistry	11

Table 6.7: Teachers information

6.6.1 Summary

The interviews with teachers revealed a shared appreciation for the role of tactile graphics in improving the learning experience for visually impaired students, particularly in STEM subjects.

However, they also highlighted the significant challenges associated with their widespread implementation, including the time-consuming and demanding process of creation, the limited availability of resources, and the difficulties some students experience with tactile perception. Teachers expressed great enthusiasm for the TAURIS app, believing its real-time audio descriptions, interactive features, and two-handed accessibility could address these challenges. They also offered valuable suggestions for improvement, such as expanding the library of tactile graphics and adding customisation options for audio descriptions.

6.7 Discussion

6.7.1 Assistive Technology Performance and Impact

This study evaluated the efficacy of smartphone-based assistive technology in enhancing the accessibility of TG within educational contexts. Quantitative analyses indicated that the technology significantly reduced the time required to acquire information compared to traditional methods such as Braille text and screen readers. This improvement can be attributed to real-time audio descriptions that enable simultaneous tactile exploration and auditory feedback. Additionally, the findings revealed that the integration of a multi-modal system resulted in substantially higher accuracy in information acquisition. This outcome likely arises from the direct coupling of auditory feedback with tactile interaction, thereby minimizing the reliance on text or screen reader navigation.

Moreover, the study investigated the impact of real-time speech output on memory retention, identifying statistically significant improvements in memory accuracy scores when using multimodal approaches as opposed to screen readers. This suggests that combining tactile exploration with spatially connected audio descriptions enhances learning experiences by fostering greater engagement and retention. While screen readers provide a linear, auditory presentation of information, the app's ability to directly link audio cues to specific tactile elements may promote a more engaging and memorable learning experience. These findings underscore the potential of mobile assistive technologies to empower visually impaired users by enabling independent and effective interaction with educational materials.

Previous studies demonstrated that TGs on their own are not sufficient to aid visually impaired people in learning (Zebehazy and Wilton, 2014c; Sheppard and Aldrich, 2001; Zebehazy and Wilton, 2014b). Experiments conducted in the research revealed that audio information from the TAURIS app enhanced the value and content of the tactile images. This finding is consistent with those of Melfi et al. (2020) and Fusco and Morash (2015). Contrary to expectations, this study did not find statistically significant differences between different vision loss and age groups.

6.7.2 Qualitative Insights and User Experiences

Quantitative findings demonstrate the potential of smartphone-based assistive technology to improve efficiency and accuracy in TG exploration. Qualitative data, including teacher observations and user feedback, highlights the impact of these technologies on learning and understanding. Participants reported a newfound clarity and confidence in explaining the frog lifecycle and the map of Australia respectively, attributing this improvement to the app's simultaneous audio and tactile feedback. This suggests that the multi-modal approach may facilitate a deeper understanding of concepts, allowing users to better connect tactile elements with their associated descriptions. However, one participant's feedback regarding the space shuttle graphic, where they felt finer details were difficult to convey due to the limited tactile exploration area, highlights a potential area for improvement.

This need for continued development is further echoed in the perspectives shared by teachers during the qualitative interviews. While acknowledging the inherent value of tactile graphics, particularly for conveying visual and spatial concepts in STEM subjects, teachers also emphasised the challenges associated with their current use. The time-consuming creation process, the limited availability of pre-made resources, and student difficulties with tactile perception were identified as key barriers to wider tactile graphic integration. Echoing the positive feedback from users, teachers expressed optimism that the TAURIS app, with its features like real-time audio descriptions, two-handed exploration, and potential for future enhancements, could effectively address these challenges. They envisioned the app as a tool to enhance understanding, promote engagement, and support individual learning needs, suggesting that further research focus on expanding the library of available graphics and providing additional customisation options.

6.7.3 Camera Aiming Improvements

The successful implementation of ARUCO markers and vibration feedback in the TAURIS app demonstrates a significant improvement in camera aiming for Visually Impaired People (VIP). Prior to the experiments, interviews revealed that the users had faced consistent difficulties when aiming phone cameras for daily use (Section 6.3.2). This echoes with the results of previous studies where participants who use a phone camera expressed difficulty in performing this task in their daily lives (Section 2.2.3). However, the results of this study showed that 12 out of 20 participants reported improved ease in camera aiming using the proposed system. More importantly, a 100% success rate in correctly aligning the camera with TG during testing. The use of ARUCO markers provided a consistent spatial reference for camera orientation, addressing the challenge of accurately positioning the camera. Furthermore, the implemented real-time vibration feedback served as a non-visual cue for users, prompting them to make the necessary adjustments for proper aiming. These results align with previous research that stresses the need for active feedback and clear spatial cues to aid visually impaired users in the task of

camera aiming (Vázquez and Steinfeld, 2012). Unlike existing solutions that are often reliant on speech only or manual guidance, the TAURIS system coupled multiple methods resulting in improved precision and a reduction in the cognitive load needed for camera positioning. The positive feedback from participants about the markers and vibration providing them confidence, also supports a more intuitive and reliable process, demonstrating the importance of incorporating user-centered design principles for practical and accessible solutions. It is therefore safe to conclude that the proposed system offers a promising approach to addressing the long-standing problem of camera aiming for visually impaired individuals.

Prior to conducting the experiments, the participants were instructed to mount the mobile phone on a phone holder and align its camera accurately with the TG placed in front of them. The results of the Likert-scale questions indicated that 12 out of 20 participants found it easy to align the camera, as opposed to the previous finding where all participants who use a phone camera expressed difficulty in performing this task in their daily lives (Section 2.2.3). This improvement may be attributed to the utilisation of ARUCO markers placed at the corners of the TG, which facilitated proper camera alignment. This finding, while preliminary, suggests that ARUCO markers are a valuable tool in ensuring accurate phone camera aiming.

6.7.4 Technical Aspects and Future Development

In Section 2.4.6, I review existing educational systems and outline the four key criteria that guided my selection of a method for detecting fingertips in the research. First of all, the developed algorithm has to be highly accurate and able to perform well in low-light conditions. Second, it was crucial to implement the whole system on a mobile device. The results of previous research show that bulky and cumbersome solutions are not willingly accepted by end-users (Ducasse et al., 2016). Third, since my device is designed to assist VIP in classrooms during their course times, a real-time execution is essential. This will allow users to keep up with their sighted peers. Lastly, the system had to allow a two-handed exploration for the user. Following studies show that it is easier to recognise tactile images using both than one hand (Wijntjes et al., 2008a; Bara, 2014). Table 6.8 presents a comparison between the TAURIS fingertip detection model and algorithms used in various systems. Based on the results, it is evident that the TAURIS system's fingertip detection algorithm is the only one that meets all of the criteria

Device	**Fingertip detection method**	**Two-handed exploration**	**Mobile device**	**Real-time**	**Detection in low-light conditions**
TAURIS (2022)	Tiny-YOLOv3	✓	✓	✓	✓

Device	**Fingertip detection method**	**Two-handed exploration**	**Mobile device**	**Real-time**	**Detection in low-light conditions**
THATS[3] (2020)	Not stated	x	✓	✓	Not stated
TARS (Hosokawa, Miwa, and Hashimoto, 2020)	MediaPipe	✓	✓	x	✓
Tactile Graphics with a Voice(Baker et al., 2014)	Colour-based skin detection	x	✓	✓	x
Tactile Graphics Helper (Fusco and Morash, 2015)	Hand segmentation	x	x	✓	x
Access Lens (Kane, Frey, and Wobbrock, 2013)	Colour-based skin detection	x	x	✓	x
Tactile Graphics Reading Assistive Device (Bahrin, Yusof, and Na'im Sidek, 2022)	MediaPipe	✓	x	✓	✓
Unified gesture recognition and fingertip detection (Alam, Islam, and Rahman, 2022)	YOLO9000	✓	x	✓	✓
Airpen (Jain and Hebbalaguppe, 2019)	MobileNet V2	x	✓	✓	✓

[3]https://thats.wiki.procivic.ro/en/

Device	Fingertip detection method	Two-handed exploration	Mobile device	Real-time	Detection in low-light conditions

Table 6.8: Fingertip detection methods comparison

Participant feedback will be used to improve the app in the future. The improvements will include adding a customisation option to increase the speed of speech output, change the synthetic voice and notifying users when light levels are too low and the algorithm is struggling to detect the fingertips. Also, it should be useful to create a universal frame with built-in ARUCO markers to increase the effective area of the printed tactile images.

During the interviews some teachers mentioned that it is challenging to find high-quality and ready to use TG. In section 4.3, I listed various repositories containing TG, but they do not fully meet the needs of the school curriculum. This is an important problem that needs more study. In fact, in a survey conducted by Sheppard and Aldrich (2001), teachers of visually impaired students say that the laborious nature of creating TG is the main challenge. My potential solution is to establish a collaborative database where teachers can share their graphics, but this will require time to build up content. My future plan is to use AI to automate the TG creation process. Generative Adversarial Networks (GANs) are a type of artificial intelligence model that are used to generate synthetic data, that are similar to real data. They do this by training two neural networks, one to generate data and one to determine if the generated data is real or synthetic. This can be used to transform visual images to tactile ones by training a GAN on a dataset of both real images and their tactile representations. To the best of my knowledge, no such AI model exists, making this a promising research direction.

Another direction of future research is to improve the existing fingertip detection model. This can be achieved by collecting additional images from individuals of diverse ages and skin tones. This will increase the model's generalisability, allowing it to be utilised by a wider population. The current system has not been tested on individuals with diverse skin tones, highlighting the need for improvement. As previously mentioned in the introduction, the developed system is open-source and I will make the object detection model and app source code accessible to other researchers.

Several participants expressed a preference for interacting with the app in the Kazakh language, highlighting the importance of language accessibility within the educational context. To address this need, a Kazakh language text-to-speech (TTS) synthesizer was integrated into the TAURIS app. While a native Kazakh TTS was unavailable on Android platforms, a third-party API provided by the Scientific Center "Til-Qazyna" was successfully utilised to incorporate this functionality. However, further evaluation involving end-users is necessary to thoroughly assess the efficacy and usability of the Kazakh TTS.

The most important limitation of the study is the small sample size. On the other hand, this is not uncommon in research which involves individuals with disabilities. Therefore, there would be value in carrying out the same experiments with a larger number of participants in several different schools for the blind and also different countries in order to obtain further useful insights, increase the power of the statistical analysis and enable further comparisons.

6.8 Conclusion

This chapter presents and discusses the key findings of the study, which evaluated the effectiveness of the TAURIS app in facilitating TG exploration. The results of end-user testing, incorporating both quantitative metrics and qualitative insights, highlight the potential of smartphone-based Assistive Technology (AT) to enhance information accessibility from TG. Specifically, the research investigated the app's influence on user perceptions (RQ1), the role of real-time speech output in improving comprehension and retention (RQ2), and approaches to optimize camera aiming (RQ3).

The findings demonstrate that the TAURIS app enables users to interact with TG more efficiently and delivers an enhanced user experience. The integration of Likert-scale responses with open-ended interview data provided a deeper understanding of user preferences and perceptions, as well as the effectiveness of real-time speech output in supporting comprehension and memory retention. Additionally, the study identified the combined use of ARUCO markers and vibration feedback as a highly effective method for significantly improving camera aiming. By employing a joint display method, the research triangulated qualitative and quantitative data, thereby reinforcing the validity of the results and offering a comprehensive evaluation of the TAURIS app's efficacy.

To gain a deeper understanding of the app's potential impact within the educational setting, interviews were conducted with teachers of visually impaired students. This qualitative data provided valuable insights into the challenges associated with current tactile graphic use, as well as the potential benefits of the TAURIS app in addressing these challenges. Teachers recognised the app's ability to enhance student understanding, engagement, and independence, offering valuable suggestions for future development.

Overall, the results of this study contribute to the understanding of the potential of the TAURIS application as an educational tool, and provide insight for future research in this area.

Chapter 7

Conclusions

This thesis consists of seven chapters, each exploring a specific aspect of making Tactile Graphics (TG) more accessible for Visually Impaired People (VIP). Chapter 2 provides a comprehensive review of the relevant literature, examining the significance of TG in Assistive Technology (AT) and identifying the barriers that have impeded their widespread utilisation. It also presents a detailed analysis of existing fingertip detection and tracking methods, revealing their limitations in the context of real-time applications on mobile devices. Chapter 3 introduces a novel fingertip detection algorithm, specifically tailored to address the challenges identified in the literature review. Chapter 4 presents a thorough overview of the developed TAURIS system — a novel, mobile-based system designed to provide real-time audio descriptions of TG, enabling independent exploration — covering its core components: the mobile application, the web-based annotation tool, and the pre-labelled TG. Chapter 5 outlines the research methodology employed in this study, encompassing the research questions, data collection methods and analysis procedures. Chapter 6 presents the results of the quantitative experiments and qualitative interviews, investigating the effectiveness and usability of the TAURIS app compared to traditional methods. Finally, Chapter 7 offers a comprehensive conclusion to the study, summarising the outcomes for each research question, outlining the study's contributions and acknowledging its limitations. Lastly, I discuss potential areas for future research, building upon the themes that emerged throughout the thesis.

7.1 Thesis overview

Chapter 2 provides a comprehensive literature review, establishing the current state of the field regarding TG and their application in AT for VIP. This review reveals key gaps in the existing literature. First, while TG are recognised as valuable tools for VIP, their wider utilisation is impeded by factors like labor-intensive production, high costs, and the declining Braille literacy among learners. Second, a review of fingertip detection and tracking methods revealed limitations in both classic Computer Vision (CV) and Deep Learning (DL) approaches. Classic CV

methods are often sensitive to variations in lighting and cluttered backgrounds, while existing DL models tend to be computationally intensive, struggling to operate in real-time on mobile devices. Finally, an examination of existing educational systems for VIP highlights the lack of solutions that effectively address these limitations, particularly those running on readily available mobile devices. The development of the TAURIS system was driven by these identified gaps, aiming to create a more accessible, efficient, and portable solution for TG exploration. The comparison of TAURIS to existing systems is presented in Table 6.8.

Chapter 3 introduces the novel fingertip detection method developed for the TAURIS app, which will be presented in Chapter 4. First, this chapter describes the development process of the fingertip detection model used in the TAURIS system. Through experimentation with three different Convolutional Neural Network (CNN) architectures, the tiny-YOLOv3 algorithm was selected due to its promising inference speed, even though initial accuracy results were not the highest. To improve accuracy, the existing dataset was expanded and the model was retrained. The results of these efforts, as shown in Table 3.3, demonstrated improved accuracy while maintaining fast detection speed. Further testing under various lighting conditions confirmed the robust performance of the model. To further enhance accuracy, "all or nothing", median filter and Kalman filter algorithms were applied, making the system more robust.

Chapter 4 presents a comprehensive overview of the TAURIS system, a novel solution designed to enhance the accessibility of TG for VIP by providing real-time audio descriptions. The chapter also covers the key components of the system, including its mobile app and web-based annotation tool. The chapter highlights the incorporation of a fingertip detection model into the app, as well as the use of ARUCO markers and QR code detection libraries for improved functionality. The utilisation of ARUCO markers aids the user in accurately aiming a phone camera, while QR codes enable the downloading of information about graphics from a server. The chapter also outlines the various description modes and mapping algorithms utilised by the TAURIS system. Finally, it provides a brief overview of the design of the annotation tool.

Chapter 5 details the research methodology employed in this study. It begins by introducing the research questions that guided the investigation and then explains the rationale behind the chosen methodological design. This study utilises a convergent mixed-methods approach, leveraging both quantitative and qualitative data to gain a comprehensive understanding of the effectiveness and user experience of the TAURIS system. The specific methods used for data collection are described in detail, including the experimental design employed to assess the app's performance compared to traditional methods. Additionally, the semi-structured interview procedures were designed to gather in-depth feedback from visually impaired participants and teachers of the visually impaired. Finally, a six-step process for data analysis is outlined.

Chapter 6 presents the results and discussions of the research, drawing upon data gathered from a diverse group of 20 participants. This group includes VIP of varying ages, vision levels, and educational backgrounds. The results of the device testing are then presented. These include

task completion times and accuracy scores, as well as their significance. The non-parametric statistical approach was employed due to the non-normality of the data. The results show that the TAURIS app allows users to acquire information from TG more efficiently. This finding is corroborated by post-experiment questionnaires and interviews, which indicate a positive user perception of the app's effectiveness and usability. A mixed-methods approach, merging and triangulating both quantitative and qualitative data, provides a comprehensive understanding of the app's impact, confirming its potential to enhance TG accessibility and support independent learning. Based on user feedback, suggestions for future work include incorporating features like adjustable speech speed, enhanced camera aiming guidance, and expansion of the TG library with more diverse content.

7.2 Research Questions

To investigate the three research questions guiding this study, a mixed-methods approach was employed, combining both quantitative and qualitative data collection and analysis techniques. During the quantitative phase of the research, participants engaged in a series of timed trials, exploring TG using the TAURIS app, Braille text, and screen reader modes. Data on task completion time, accuracy in answering questions, and memory recall were recorded for each participant and each exploration mode. This data was then analysed using non-parametric statistical methods, specifically the Wilcoxon signed-rank test for within-subject comparisons and the Mann-Whitney U test for between-subject comparisons, to assess the significance of the observed differences. Qualitative data was gathered through semi-structured interviews conducted with each participant, exploring their perceptions, experiences, and suggestions regarding the TAURIS app. These interviews were audio-recorded, transcribed verbatim, and thematically analysed using a deductive coding approach to identify key themes and patterns in the participant feedback.

7.2.1 Research Question 1

- **RQ1:** *What are visually impaired individuals' **perceptions and attitudes** toward the use of smartphone app in the context of exploring Tactile Graphics (TG)?*

This study reveals a positive perception among VIP towards using the TAURIS app for TG exploration. The combined analysis of Likert-scale data and qualitative interviews shows a strong preference for the TAURIS app, with a significant majority of participants (90%) expressing interest in daily use. Highlighting the app's superior convenience, speed, and engagement when compared to traditional Braille text and screen readers. The positive sentiment was further supported by the detailed qualitative analysis, in which participants frequently described the app as "convenient," "helpful," "intuitive," and "interactive." They valued the app's ability to

provide access to information and reduce the cumbersome back-and-forth navigation associated with traditional methods. These insights, coupled with valuable recommendations for future improvements, particularly regarding the need for customisation and expanded content options, emphasize the TAURIS app's effectiveness in enhancing the accessibility and usability of TG. The feedback from participants also highlighted specific design elements, such as real-time audio descriptions, that users considered to be important for creating user-friendly and effective AT solutions for TG exploration. While the overall response to the TAURIS app is largely positive, it's important to acknowledge the limitations of this study, including the small sample size which might limit generalisation of these findings. However, the consistent feedback and preferences indicate a strong foundation for future design and development efforts.

7.2.2 Research Question 2

- **RQ2:** *To what extent does real-time speech output, integrated with tactile exploration, enhance the* ***comprehension and retention*** *of complex information conveyed through TG for visually impaired users?*

This study demonstrates that the integration of real-time, spatially-connected audio descriptions with tactile exploration, as implemented in the TAURIS app, significantly enhances the comprehension and retention of complex information for VIP. Quantitative results showed that participants using the TAURIS app achieved significantly faster response times and higher accuracy compared to both Braille text and screen reader modes. Furthermore, memory recall was also significantly higher compared to screen reader mode. These statistically significant improvements, supported by the qualitative feedback, highlight the effectiveness of the app's multi-modal approach which combines tactile engagement with corresponding real-time audio descriptions. The qualitative data further supported these findings, with participants reporting a confidence in understanding complex information and improved spatial understanding of concepts, such as the frog lifecycle and maps. Specifically, the app was noted to enable the creation of a mental 2D representation of the data, which aided in information recall. While the TAURIS app showed great promise in aiding information processing, it must be acknowledged that the effectiveness of the app is dependent on factors such as the design of the TG itself. Smaller and overly complex graphics may not benefit as much from real-time audio description as simpler ones. In addition, it must be recognized that the scope of this study only focused on a few specific types of TGs. As such, this highlights the need for careful consideration of the TG design and complexity to enable the optimal impact of real-time audio feedback. While this study's results indicate that the app can improve information processing through multi-modal interfaces for visually impaired learners, further research will be essential to fully validate the effectiveness and impact on learning in different contexts.

7.2.3 Research Question 3

- **RQ3:** *What methods can be employed to improve **camera aiming** in smartphone-based assistive technology applications designed for exploring TG?*

This study identified ARUCO markers as highly effective for improving camera aiming for visually impaired users. When combined with vibration feedback, these markers enhanced interactions with smartphone-based AT for TG exploration. The use of ARUCO markers, strategically placed at the corners of TG, provided a reliable spatial reference that enabled a 100% success rate in proper alignment during user testing. Furthermore, the integration of real-time vibration feedback facilitated a non-visual method for users to correct misalignments, providing immediate tactile cues for optimal positioning. These findings highlight the benefit of combining multiple approaches to enhance user experience and system accuracy in camera aiming. Subjective user feedback collected during interviews supported the effectiveness of these methods, demonstrating an increased confidence that the camera was aligned properly with the TG. Importantly, while these results indicate a positive outcome, further development is needed to create a more flexible system that reduces the reliance on pre-printed graphics and provides more explicit audio navigation for precise adjustments. Therefore, it can be concluded that while the TAURIS system has addressed a significant challenge in using smartphone cameras for TG, a continued focus on exploration of various camera aiming methodologies is required for broader applications.

7.3 Contributions

In this section I summarise the main contributions of my work on the development of AT for VIP.

1. **System to Provide Real-time Audio Descriptions for Tactile Graphics.** This thesis contributes a novel system designed to provide VIP with real-time audio descriptions of TG, enabling them to access and understand graphical information independently, without requiring sighted assistance. Unlike existing solutions that often rely on bulky hardware, pre-recorded audio, or limited single-finger exploration, the TAURIS system leverages a smartphone's camera and processing capabilities to offer a more portable, interactive, and comprehensive experience. The system incorporates a customised DL model for accurate fingertip detection, even under varying lighting conditions, and enables two-handed exploration, allowing users to engage with TG more naturally. To evaluate the system's usability, user testing was conducted with a diverse group of VIP, focusing on metrics such as task completion time, accuracy in answering questions, and user perceptions obtained through questionnaires and semi-structured interviews. These findings provided valuable insights into the system's strengths, weaknesses, and potential areas for improvement.

2. **Fast and accurate fingertip detection model.** This thesis contributes a novel fingertip detection model that is both highly accurate and capable of real-time operation on a mobile device, a significant advancement in the field of TG exploration for VIP. The model, presented in Chapter 3, achieves high accuracy through a combination of a customised dataset tailored specifically for TG exploration and a YOLOv3 DL model selected for its efficiency and speed. This model successfully detects all ten fingertips simultaneously, enabling two-handed exploration, which previous models have struggled to achieve in real-time. Two-handed exploration allows for more natural and intuitive interactions with TG, as users can simultaneously trace contours, explore spatial relationships between elements, and maintain a better overall understanding of the graphic's layout. While some existing systems offer two-handed interaction with tactile displays or 3D models, to the best of my knowledge, this is the first system to achieve real-time, accurate fingertip detection for two-handed exploration on standard TG using a mobile device.

3. **Novel fingertip detection image set.** This thesis introduces a novel dataset specifically designed for training and evaluating fingertip detection models on TG. This dataset, consisting of 2000 manually annotated images captured during real-world TG exploration sessions, offers several advantages over existing datasets primarily focused on hand gestures or object recognition in general scenes. The images in this dataset capture the unique challenges of fingertip detection on TG, such as variations in lighting, hand positions, and occlusions caused by interaction with raised lines and textures. This dataset significantly improved the accuracy of the developed fingertip detection model, with an average increase of 20% for each finger. By making this dataset publicly available under a Creative Commons license, this thesis aims to accelerate research in this field, enabling other researchers to benefit from this resource and further advance the development of accessible TG exploration technologies.

4. **Tests under different lighting conditions.** This thesis demonstrates the robustness of the developed fingertip detection model under a wide range of lighting conditions, a crucial aspect for ensuring the usability and reliability of AT for VIP. While previous research in gesture and fingertip detection has often focused on controlled lighting environments, this study explicitly evaluated the model's performance under varying illumination levels. As described in Chapter 3, a dedicated test set of 200 images was created, and the gamma correction algorithm was utilised to alter image brightness, simulating different lighting conditions encountered in real-world scenarios. The results, summarised in Table 3.5, demonstrate that the TAURIS system maintains a high level of accuracy even in low-light conditions, surpassing the performance of several existing systems designed for gesture recognition. This finding constitutes a significant contribution to the field, providing a valuable reference for future researchers and developers aiming to create robust and reli-

able assistive technologies that perform effectively across diverse lighting environments.

5. **Enhancing App Performance and Reliability Through Advanced Algorithms.** This thesis contributes to the field of accessible TG exploration by developing and integrating a combination of algorithms designed to enhance the performance and reliability of the TAURIS app. These algorithms, implemented in the app's fingertip detection and tracking pipeline, go beyond simply applying existing techniques. They represent modifications and novel approaches specifically tailored to the unique challenges of real-time fingertip detection for VIP.

    - **"All or Nothing" Approach:** To minimise false positive detections, a novel "all or nothing" algorithm was implemented. This algorithm ensures that only when all fingers, except the thumb, are confidently detected by the deep learning model will the location of the index finger be used for triggering audio descriptions. This approach significantly reduces erroneous audio cues caused by misdetected fingers.
    - **Median Filter:** To address instances where the model misidentifies the same finger on both hands, a median filter was developed and integrated. This filter smooths the raw fingertip location data by considering a window of consecutive frames, effectively reducing erroneous jumps between left and right hands.
    - **Kalman Filter:** To further enhance tracking stability and mitigate the impact of spurious or missing detections, a Kalman filter was implemented. By leveraging temporal information from consecutive video frames, the Kalman filter predicts the future location of the fingertip, effectively smoothing its trajectory and ensuring that audio feedback remains consistent.

    These algorithms, operating in conjunction, substantially improve the overall performance and reliability of the TAURIS system, providing users with a smoother, more accurate, and less error-prone TG exploration experience.

6. **Enhanced Camera Aiming Using ARUCO Markers and Vibration Feedback** This thesis introduces a novel approach to improving camera aiming accuracy for VIP, a persistent challenge in mobile AT. The TAURIS system utilises ARUCO markers placed at the corners of TG to provide spatial references for the phone camera. By detecting these markers, the app can determine if the camera is properly aligned and the entire graphic is within the field of view. The system incorporates vibration feedback to alert users when one or more markers are not visible, prompting them to readjust the phone's position. Each marker has a unique identifier, allowing the app to determine which corner is missing and guide the user with more specific directional cues. This approach has significantly improved camera aiming accuracy during user testing, with all participants achieving a 100% success rate in properly aligning the camera with the TG. While a voice assistant

feature that provides explicit verbal guidance is not yet implemented, it is planned for future development to further enhance usability. The challenge of accurate camera aiming remains a major area of focus in the field of AT, and this thesis offers a promising solution for visually impaired users interacting with TG on mobile devices.

7. **Addition of Kazakh language** This thesis addresses a critical gap in accessible learning resources by integrating Kazakh language support into the TAURIS system. While numerous AT and learning materials are readily available in English and Russian, there is a significant lack of resources for Kazakh speakers, particularly in the domain of TG exploration. This was achieved by incorporating a recently released and publicly available Kazakh Text-To-Speech (TTS) engine, enabling the delivery of all audio descriptions and app feedback in Kazakh, in addition to the existing English and Russian language support. This localised language functionality makes the system readily accessible to a wider population of VIP in Kazakhstan, where Kazakh is the primary language. By demonstrating the feasibility of integrating a Kazakh TTS engine, this thesis sets a precedent for making AT available in languages for which there is currently limited provision. This contribution highlights the importance of considering linguistic diversity and promoting inclusion in the design and development of AT, ensuring that they reach and benefit a truly global community of users.

7.4 Strengths and Limitations

This section reflects upon both the strengths and limitations of the research, providing a transparent assessment of its contributions and highlighting areas for future development.

Strengths

- **Novel and Accessible System:** The TAURIS system represents a significant advance in TG exploration for VIP. Its unique combination of real-time fingertip detection, two-handed interaction, and customisable audio descriptions offers a more engaging, intuitive, and independent learning experience compared to traditional methods.
- **Robust Fingertip Detection:** The developed fingertip detection model, incorporating a customised dataset and a YOLOv3 DL model, achieves high accuracy even under varying lighting conditions, addressing a significant challenge in Computer Vision-based AT.
- **Accessibility for Non-Russian Speakers:** The integration of Kazakh language support through a TTS engine demonstrates the feasibility and importance of extending AT to under-resourced languages, promoting inclusion and broadening the system's reach.

- **Enhanced Camera Aiming:** The use of ARUCO markers and vibration feedback effectively guides visually impaired users in properly aligning the phone camera, improving the accuracy and usability of the system.

Limitations

- **Small Sample Size:** The limited sample size in the user testing, primarily due to unforeseen circumstances caused by the COVID-19 pandemic, may impact the generalisability of the findings. Further research with a larger and more diverse participant pool is needed to strengthen the study's conclusions.
- **Focus on Two-Handed Exploration:** The study primarily focused on two-handed TG exploration, as participants found this approach more convenient. Further research is needed to evaluate the app's effectiveness and usability for individuals who prefer or require one-handed exploration.
- **Limited Skin Tone Diversity:** While the fingertip detection model performed well during user testing, its accuracy on a wider population with diverse skin tones has not been extensively evaluated. Further research and dataset expansion are necessary to ensure the model's robustness and generalisability across different skin tones.

7.5 Future Work

A natural progression of this work is to replicate the study in various educational settings with a larger sample size. This would provide a more comprehensive understanding of the findings and generalisability of the results. Additionally, a comparative study to analyse the performance and user experience of one-handed and two-handed TG exploration with the TAURIS app, would provide valuable insights into the optimal interaction modes for different users and learning contexts.

Also further work will have to be conducted on refining the existing app based on user feedback and suggestions. This includes the implementation of a universal frame with ARUCO[1] markers placed at the corners to increase the effective area of the TG, the inclusion of a light level indicator to alert users when lighting conditions are insufficient, and the provision of a customisable speech output speed option to accommodate individual preferences. These improvements will enhance the overall experience of visually impaired users and better meet their needs.

Another promising avenue for future research is the development of automated tools for creating accessible materials. Specifically, incorporating AI technology to convert visual graphics

[1] ARUCO markers are fiducial markers that can be easily detected by computer vision algorithms, providing reliable reference points for camera pose estimation and object tracking.

into tactile representations. Furthermore, the results of this study have raised several questions regarding the role of ARUCO markers in assisting VIPs in aiming their phone cameras. Further investigation into this aspect could provide valuable insights and inform the design of future assistive technologies.

Finally, it would be beneficial to extend the fingertip detection assessment to a larger population to assess its performance and identify any demographic or individual differences that may impact the model detection results.

7.6 Final Remarks

This thesis makes a significant contribution to overcoming the barriers that visually impaired individuals face in accessing information, particularly within educational settings. By developing and evaluating the TAURIS system, this research not only provides a novel and practical tool for independent TG exploration, but also demonstrates the broader potential of mobile, multi-modal technologies in enhancing learning for VIP. Through this thesis, I explored the efficacy of the TAURIS system, which leverages a smartphone's camera, processing capabilities, and audio output to provide real-time, spatially-connected audio descriptions of TG. The results of this research demonstrate that this approach is more effective in supporting information acquisition and comprehension than traditional tools, such as Braille texts and screen readers, which often require cumbersome navigation and can disrupt the learning flow. I believe that the wider adoption and utilisation of mobile, multi-modal systems like TAURIS hold significant promise in empowering visually impaired learners to access knowledge, engage with educational materials, and pursue their academic goals in a more independent and efficient manner.

Bibliography

Abadi, Martın et al. (2016). “Tensorflow: Large-scale machine learning on heterogeneous distributed systems”. In: *arXiv preprint arXiv:1603.04467*.

Ackland, Peter, Serge Resnikoff, and Rupert Bourne (2017). “World blindness and visual impairment: despite many successes, the problem is growing”. In: *Community eye health* 30.100, p. 71.

Adams, Frank R et al. (1989). “IBM products for persons with disabilities”. In: *1989 IEEE Global Telecommunications Conference and Exhibition’Communications Technology for the 1990s and Beyond’*. IEEE, pp. 980–984.

Alam, Mohammad Mahmudul, Mohammad Tariqul Islam, and SM Mahbubur Rahman (2022). “Unified learning approach for egocentric hand gesture recognition and fingertip detection”. In: *Pattern Recognition* 121, p. 108200.

Angelova, Anelia et al. (2015). “Real-time pedestrian detection with deep network cascades”. In.

Atlas, Iapb Vision (2020). “Magnitude and Projections”. In: *Int. Agency Prev. Blind.[Online]. Available: https://www.iapb.org/learn/vision-atlas/magnitude-and-projections/countries/kazakhstan/. 14-Feb-2023]*.

Ayala-Ramirez, Victor et al. (2011). “A hand gesture recognition system based on geometric features and color information for human computer interaction tasks”. In: *The Journal of Pattern Recognition Society*, pp. 54–59.

Bahrin, Muhammad Ikmal Hakim, Hazlina Md Yusof, and Shahrul Na’im Sidek (2019). “Tactile Graphics Exploration Studies Using Fingertip Tracking Based on Colour Markers Detection for Visually Impaired People”. In: *2019 7th International Conference on Mechatronics Engineering (ICOM)*. IEEE, pp. 1–6.

Bahrin, Muhammad Ikmal Hakim Shamsul, Hazlina Md Yusof, Shahrul Na’im Sidek, and Aimi Shazwani Ghazali (2024). “An Early Investigation into Raised-Line Tactile Graphics Reading Behavior among Blind and Visually Impaired Individuals”. In: *Jurnal Kejuruteraan* 36.3, pp. 1103–1125.

Bahrin, Yusof, and Shahrul Na’im Sidek (2022). “Hands and Fingers Tracking for Tactile Graphics Reading Assistive Device”. In: *Enabling Industry 4.0 through Advances in Mechatronics: Selected Articles from iM3F 2021, Malaysia* 900, p. 413.

Bai, Junjie, Lu, and Zhang (2019). *ONNX: Open Neural Network Exchange*. `https://github.com/onnx/onnx`.

Baker, Catherine M et al. (2014). "Tactile graphics with a voice: using QR codes to access text in tactile graphics". In: *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility*, pp. 75–82.

Balata, Jan, Zdenek Mikovec, and Lukas Neoproud (2015). "Blindcamera: Central and golden-ratio composition for blind photographers". In: *Proceedings of the Mulitimedia, Interaction, Design and Innnovation*, pp. 1–8.

Baldauf, Matthias et al. (2011). "Markerless visual fingertip detection for natural mobile device interaction". In: *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp. 539–544.

Bara, Florence (2014). "Exploratory procedures employed by visually impaired children during joint book reading". In: *Journal of developmental and physical disabilities* 26.2, pp. 151–170.

Bardot, Sandra et al. (2017). "Identifying how visually impaired people explore raised-line diagrams to improve the design of touch interfaces". In: *Proceedings of the 2017 CHI conference on human factors in computing systems*, pp. 550–555.

Beck, Cheryl Tatano, Carrie Morgan Eaton, and Robert K Gable (2016). "Vicarious posttraumatic growth in labor and delivery nurses". In: *Journal of Obstetric, Gynecologic & Neonatal Nursing* 45.6, pp. 801–812.

Berla, Edward P (1972). "Effects of physical size and complexity on tactual discrimination of blind children". In: *Exceptional children* 39.2, pp. 120–124.

Berla, Edward P and Lawrence H Butterfield Jr (1977). "Tactual distinctive features analysis: Training blind students in shape recognition and in locating shapes on a map". In: *The Journal of Special Education* 11.3, pp. 335–346.

Bhuyan, MK, Debanga Raj Neog, and Mithun Kumar Kar (2012). "Fingertip detection for hand pose recognition". In: *International Journal on Computer Science and Engineering* 4.3, p. 501.

Bigham, Jeffrey P et al. (2010). "VizWiz:: LocateIt-enabling blind people to locate objects in their environment". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*. IEEE, pp. 65–72.

Boswell, Katie and Angela Kail (2016). "VISUAL IMPAIRMENT IN SCOTLAND". In: *NPC. Available: https://www.thinknpc.org/wp-content/uploads/2018/07/Visual-Impairment-in-Scotland-a-guide-for-funders1.pdf [Accessed: 14-Feb-2023]*.

Bradski, Gary and Adrian Kaehler (2008). *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc."

Brulé, Emeline et al. (2020). "Review of Quantitative Empirical Evaluations of Technology for People with Visual Impairments". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–14.

Budrionis, Andrius et al. (2022). "Smartphone-based computer vision travelling aids for blind and visually impaired individuals: A systematic review". In: *Assistive Technology* 34.2, pp. 178–194.

Casado-Garcıa, Ángela and Jónathan Heras (2020). "Ensemble methods for object detection". In: *ECAI 2020*. IOS Press, pp. 2688–2695.

Cavazos Quero, Luis, Jorge Iranzo Bartolomé, and Jundong Cho (2021). "Accessible visual artworks for blind and visually impaired people: comparing a multimodal approach with tactile graphics". In: *Electronics* 10.3, p. 297.

Chan, T, Y Yu, and K Wong (2018). "Text detection and marker based finger tracking in building a language assistant for wearable glasses". In: *International Conference on Internet Studies*.

Chen, Mu Li, et al. (2015). "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems". In: *arXiv preprint arXiv:1512.01274*.

Chen, Zhuo, Xiaoming Liu, et al. (2021). "A wearable navigation device for visually impaired people based on the real-time semantic visual slam system". In: *Sensors* 21.4, p. 1536.

Chollet, François et al. (2015). *keras*.

QR-code-generator.com (2020). *Getting Started With QR Codes*. URL: `https://www.qr-code-generator.com/qr-code-marketing`.

Creswell, John W and Vicki L Plano Clark (2017). *Designing and conducting mixed methods research*. Sage publications.

Creswell, John W, Vicki L Plano Clark, et al. (2003). "Advanced mixed methods research designs". In: *Handbook of mixed methods in social and behavioral research* 209.240, pp. 209–240.

Dandekar, Kiran, Balasundar I Raju, and Mandayam A Srinivasan (2003). "3-D finite-element models of human and monkey fingertips to investigate the mechanics of tactile sense". In: *J. Biomech. Eng.* 125.5, pp. 682–691.

Deng, Jia et al. (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

Denscombe, Martyn (2008). "Communities of practice: A research paradigm for the mixed methods approach". In: *Journal of mixed methods research* 2.3, pp. 270–283.

— (2017). *EBOOK: The good research guide: For small-scale social research projects*. McGraw-Hill Education (UK).

Dias, M Bernardine et al. (2010). "Experiences with lower-cost access to tactile graphics in India". In: *Proceedings of the First ACM Symposium on Computing for Development*, pp. 1–9.

Duan, Haihan et al. (2020). "Ambient light based hand gesture recognition enabled by recurrent neural network". In: *IEEE Access* 8, pp. 7303–7312.

Ducasse, Julie et al. (2016). "Tangible reels: construction and exploration of tangible maps by visually impaired users". In: *Proceedings of the 2016 CHI conference on human factors in computing systems*, pp. 2186–2197.

Everingham, Mark et al. (2010). "The pascal visual object classes (voc) challenge". In: *International journal of computer vision* 88.2, pp. 303–338.

Fang, Yikai et al. (2007). "A real-time hand gesture recognition method". In: *2007 IEEE International Conference on Multimedia and Expo*. IEEE, pp. 995–998.

Feiz, Shirin et al. (2019). "Towards enabling blind people to independently write on printed forms". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–12.

Fukushima, Kunihiko and Sei Miyake (1982). "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition". In: *Competition and cooperation in neural nets*. Springer, pp. 267–285.

Fusco, Giovanni and Valerie S Morash (2015). "The tactile graphics helper: providing audio clarification for tactile graphics using machine vision". In: *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*, pp. 97–106.

Galdran, Adrian et al. (2017). "Data-driven color augmentation techniques for deep skin image analysis". In: *arXiv preprint arXiv:1703.03702*.

Gardner, John A and Vladimir Bulatov (2006). "Scientific diagrams made easy with IVEO TM". In: *International Conference on Computers for Handicapped Persons*. Springer, pp. 1243–1250.

Gemperle, Francine, Nathan Ota, and Dan Siewiorek (2001). "Design of a wearable tactile display". In: *Proceedings Fifth International Symposium on Wearable Computers*. IEEE, pp. 5–12.

Girshick, Ross (2015). "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448.

Girshick, Ross et al. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

Grady, Cheryl L et al. (1998). "Neural correlates of the episodic encoding of pictures and words". In: *Proceedings of the National Academy of Sciences* 95.5, pp. 2703–2708.

Griffin-Shirley, Nora et al. (2017). "A survey on the use of mobile applications for people who are visually impaired". In: *Journal of Visual Impairment & Blindness* 111.4, pp. 307–323.

Guerreiro, Tiago et al. (2015). "Blind people interacting with large touch surfaces: Strategies for one-handed and two-handed exploration". In: *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*, pp. 25–34.

Gurav, Ruchi M and Premanand K Kadbe (2015). "Vision based hand gesture recognition with haar classifier and AdaBoost algorithm". In: *Int J Latest Trends Eng Technol (IJLTET)* 5.2, pp. 155–160.

Harris, Charles R. et al. (Sept. 2020). "Array programming with NumPy". In: *Nature* 585.7825, pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

Hasan, Mokhtar M and Pramod K Mishra (2012). "Hand gesture modeling and recognition using geometric features: a review". In: *Canadian journal on image processing and computer vision* 3.1, pp. 12–26.

Heap, Tony and David Hogg (1996). "Towards 3D hand tracking using a deformable model". In: *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*. Ieee, pp. 140–145.

Hersh, Marion and Johnson (2008). *Assistive technology for visually impaired and blind people*. Vol. 1. Springer.

Hosokawa, Yoichi, Tetsushi Miwa, and Yoshihiro Hashimoto (2020). "Development of TARS Mobile App with Deep Fingertip Detector for the Visually Impaired". In: *International Conference on Computers Helping People with Special Needs*. Springer, pp. 435–445.

Howard, Andrew G et al. (2017). "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861*.

Institute, Braille (2010). *Facts about sight loss and definitions of blindness*. URL: http://www.brailleinstitute.org/facts_about_sight_loss#5.

Iwamura, Masakazu et al. (2020). "VisPhoto: photography for people with visual impairment as post-production of omni-directional camera image". In: *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–9.

Jain, Varun and Ramya Hebbalaguppe (Apr. 2019). "AirPen: A Touchless Fingertip Based Gestural Interface for Smartphones and Head-Mounted Devices". In.

Jayant, Chandrika et al. (2011). "Supporting blind photography". In: *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, pp. 203–210.

Jia, Yangqing et al. (2014). "Caffe: Convolutional architecture for fast feature embedding". In: *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 675–678.

Johnson (1996). "The braille literacy crisis for children". In: *Journal of Visual Impairment & Blindness* 90.3, pp. 276–278.

Johnson, Peter W and Janet M Blackstone (2007). "Children and gender—differences in exposure and how anthropometric differences can be incorporated into the design of computer input devices". In: *SJWEH Supplements* 3, pp. 26–32.

Joseph, V Roshan (2022). "Optimal ratio for data splitting". In: *Statistical Analysis and Data Mining: The ASA Data Science Journal*.

Kachouane, M et al. (2012). "HOG based fast human detection". In: *2012 24th International Conference on Microelectronics (ICM)*. IEEE, pp. 1–4.

Kane, Shaun K, Brian Frey, and Jacob O Wobbrock (2013). "Access lens: a gesture-based screen reader for real-world documents". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 347–350.

Kim, Jong-Min and Lee (2008). "Hand shape recognition using fingertips". In: *2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery*. Vol. 4. IEEE, pp. 44–48.

Kimchi, Judith, Barbara Polivka, and Joanne Sabol Stevenson (1991). "Triangulation: operational definitions". In: *Nursing research* 40.6, pp. 364–366.

Kounavis, Michael (2017). "Fingertip detection without the use of depth data, color information, or large training data sets". In: *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, pp. 2396–2401.

Krasin, Ivan et al. (2017). "Openimages: A public dataset for large-scale multi-label and multi-class image classification". In: *Dataset available from https://github. com/openimages* 2.3, p. 18.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25, pp. 1097–1105.

La Gorce, Martin de, David J Fleet, and Nikos Paragios (2011). "Model-based 3d hand pose estimation from monocular video". In: *IEEE transactions on pattern analysis and machine intelligence* 33.9, pp. 1793–1805.

Landau, Steven and Karen Gourgey (2001). "Development of a talking tactile tablet". In: *Information Technology and Disabilities* 7.2.

Lazar, Jonathan, Jinjuan Heidi Feng, and Harry Hochheiser (2017). *Research methods in human-computer interaction*. Morgan Kaufmann, p. 30.

LeCun, Yann et al. (1989). "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4, pp. 541–551.

Lee, Kyungjun et al. (2019). "Revisiting blind photography in the context of teachable object recognizers". In: *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 83–95.

Legocki, Laurie J et al. (2015). "Clinical trialist perspectives on the ethics of adaptive clinical trials: a mixed-methods analysis". In: *BMC Medical Ethics* 16.1, pp. 1–12.

Leo, Fabrizio, Elena Cocchi, and Luca Brayda (2016). "The effect of programmable tactile displays on spatial learning skills in children and adolescents of different visual disability". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.7, pp. 861–872.

Li, Yichen et al. (2018). "Self-powered gesture recognition with ambient light". In: *Proceedings of the 31st annual ACM symposium on user interface software and technology*, pp. 595–608.

Liang, Hui et al. (2013). "Model-based hand pose estimation via spatial-temporal hand parsing and 3D fingertip localization". In: *The Visual Computer* 29.6, pp. 837–848.

Likert, Rensis (1932). "A technique for the measurement of attitudes." In: *Archives of psychology*.

Lin, Piotr Dollár, et al. (2017). "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125.

Lin, Priya Goyal, et al. (2017). "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988.

Lin, Michael Maire, et al. (2014). "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer, pp. 740–755.

Liu, Wei et al. (2016). "Ssd: Single shot multibox detector". In: *European conference on computer vision*. Springer, pp. 21–37.

Lu, Xiong et al. (2020). "Kinect-based human finger tracking method for natural haptic rendering". In: *Entertainment Computing* 33, p. 100335.

Manoharan, Samuel et al. (2019). "A smart image processing algorithm for text recognition, information extraction and vocalization for the visually challenged". In: *Journal of Innovative Image Processing (JIIP)* 1.01, pp. 31–38.

Mazumder, Joy, Laila Naznin Nahar, and Md Moin Uddin Atique (2018). "Finger gesture detection and application using hue saturation value". In: *International Journal of Image, Graphics and Signal Processing* 11.8, p. 31.

Melfi, Giuseppe et al. (2020). "Understanding what you feel: A mobile audio-tactile system for graphics used at schools with students with visual impairment". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–12.

Miller, Irene et al. (2010). "Guidelines and standards for tactile graphics". In: *The Braille Authority of North America*.

Miwa, Tetsushi et al. (2020). "TARS mobile app with deep fingertip detector for the visually impaired". In: *International conference on intelligent human systems integration*. Springer, pp. 301–306.

Morse, Janice M (1991). "Strategies for sampling". In: *Qualitative nursing research: A contemporary dialogue* 127, p. 122.

Mukherjee, Sohom et al. (2019). "Fingertip detection and tracking for recognition of air-writing in videos". In: *Expert Systems with Applications* 136, pp. 217–229.

Mukhiddinov and Soon-Young (2021). "A systematic literature review on the automatic creation of tactile graphics for the blind and visually impaired". In: *Processes* 9.10, p. 1726.

Nguyen, Dung Duc, Thien Cong Pham, and Jae Wook Jeon (2009). "Fingertip detection with morphology and geometric calculation". In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 1460–1465.

Nielsen, J (2005). *Time Budgets for Usability Sessions. [online] Nielsen Norman Group.* URL: https://www.nngroup.com/articles/time-budgets-for-usability-sessions/.

NNPCPK (2022). "LIST OF SPECIAL SCHOOL ORGANIZATIONS OF EDUCATION IN KAZAKHSTAN". In: *National Scientific and Practical Center for the Development of Special and Inclusive Education. Available: https://special-edu.kz/SOO/4.2.pdf [Accessed: 17-Feb-2023]*.

Novak, Joseph D and Alberto J Cañas (2008). "The theory underlying concept maps and how to construct and use them". In.

Ofcom (2017). *Access and Inclusion in 2016*.

Oka, Kenji, Yoichi Sato, and Hideki Koike (2002). "Real-time fingertip tracking and gesture recognition". In: *IEEE Computer graphics and Applications* 22.6, pp. 64–71.

Olmschenk, Greg et al. (2015). "Mobile crowd assisted navigation for the visually impaired". In: *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*. IEEE, pp. 324–327.

Paivio, Allan (2013). *Imagery and verbal processes*. Psychology Press.

Parks, Don (1988). "AN AUDIO-TACTILE TOOL FOR THE ACQUISITION, USE AND MANAGEMENT OF SPATIALLY DISTRIBUTED INFORMA-TION BY PARTIALLY SIGHTED AND BLIND PERSONS". In: *proceedings of the second international symposium on maps and graphics for visually handicapped people*, pp. 24–29.

Pasquero, Jérôme and Vincent Hayward (2003). "STReSS: A practical tactile display system with one millimeter spatial resolution and 700 Hz refresh rate". In: *Proc. Eurohaptics*. Vol. 2003, pp. 94–110.

Paszke, Adam et al. (2019). "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32.

Patton, Michael Quinn (1990). *Qualitative evaluation and research methods*. SAGE Publications, inc.

Petit, Grégory et al. (2008). "Refreshable tactile graphics applied to schoolbook illustrations for students with visual impairment". In: *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, pp. 89–96.

Petridou, Maria (2014). "Playful haptic environment for engaging visually impaired learners with geometric shapes". PhD thesis. University of Nottingham.

Prescher, Denise, Jens Bornschein, and Gerhard Weber (2014). "Production of accessible tactile graphics". In: *International Conference on Computers for Handicapped Persons*. Springer, pp. 26–33.

Qin, Zeming Li, et al. (2019). "ThunderNet: Towards real-time generic object detection on mobile devices". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6718–6727.

Qin, Shuxin, Xiaoyang Zhu, et al. (2014). "Real-time hand gesture recognition from depth images using convex shape decomposition method". In: *Journal of Signal Processing Systems* 74.1, pp. 47–58.

Redmon, Santosh Divvala, et al. (2016). "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

Redmon, Joseph (2013–2016). *Darknet: Open Source Neural Networks in C*. `http://pjreddie.com/darknet/`.

Redmon, Joseph and Ali Farhadi (2017). "YOLO9000: better, faster, stronger". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271.

— (2018). "Yolov3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767*.

Reichinger, Andreas et al. (2016). "Gesture-based interactive audio guide on tactile reliefs". In: *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 91–100.

Ren, Shaoqing et al. (2016). "Faster R-CNN: towards real-time object detection with region proposal networks". In: *IEEE transactions on pattern analysis and machine intelligence* 39.6, pp. 1137–1149.

Al-Rfou, Rami et al. (2016). "Theano: A Python framework for fast computation of mathematical expressions". In: *arXiv e-prints*, arXiv–1605.

Roberts, Lawrence G (1963). "Machine perception of three-dimensional solids". PhD thesis. Massachusetts Institute of Technology.

Romero-Ramirez, Francisco J, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer (2018). "Speeded up detection of squared fiducial markers". In: *Image and vision Computing* 76, pp. 38–47.

Rosenblum, L Penny, Li Cheng, and Carole R Beal (2018). "Teachers of students with visual impairments share experiences and advice for supporting students in understanding graphics". In: *Journal of visual impairment & blindness* 112.5, pp. 475–487.

Ross, David A and Bruce B Blasch (2000). "Evaluation of orientation interfaces for wearable computers". In: *Digest of Papers. Fourth International Symposium on Wearable Computers*. IEEE, pp. 51–58.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1986). "Learning representations by back-propagating errors". In: *nature* 323.6088, pp. 533–536.

Ryles, Ruby (1996). "The impact of braille reading skills on employment, income, education, and reading habits". In: *Journal of Visual Impairment & Blindness* 90.3, pp. 219–226.

Sakula, Alex (1998). "That the Blind May Read: The Legacy of Valentin Hauy, Charles Barbier, Louis Braille and William Moon". In: *Journal of medical biography* 6.1, pp. 21–27.

Sandler, Mark et al. (2018). "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520.

Sarkar, Arpita Ray, G Sanyal, and SJIJOCA Majumder (2013). "Hand gesture recognition systems: a survey". In: *International Journal of Computer Applications* 71.15.

Scheithauer, Mindy C and Jeffrey H Tiger (2012). "A computer-based program to teach braille reading to sighted individuals". In: *Journal of applied behavior analysis* 45.2, pp. 315–327.

Scientific, National and Kazakhstan Practical Center named after Sh. Shayakhmetov Astana (2024). *Til – kazyna*. URL: https://tilqazyna.kz/.

Seide, Frank and Agarwal (2016). "CNTK: Microsoft's open-source deep-learning toolkit". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 2135–2135.

Sekachev, Boris et al. (Aug. 2020). *opencv/cvat: v1.1.0*. Version v1.1.0. DOI: 10.5281/zenodo.4009388. URL: https://doi.org/10.5281/zenodo.4009388.

Shamsul Bahrin, Muhammad Ikmal Hakim, Hazlina Md Yusof, and Shahrul Na'im Sidek (2022). "Hands and Fingers Tracking for Tactile Graphics Reading Assistive Device". In: *Enabling Industry 4.0 through Advances in Mechatronics*. Springer, pp. 413–422.

Shapiro, Samuel Sanford and Martin B Wilk (1965). "An analysis of variance test for normality (complete samples)". In: *Biometrika* 52.3/4, pp. 591–611.

Shen, Huiying et al. (2013). "CamIO: a 3D computer vision system enabling audio/haptic interaction with physical objects by blind users". In: *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 1–2.

Sheppard, Linda and Frances K Aldrich (2001). "Tactile graphics in school education: perspectives from teachers". In: *British Journal of Visual Impairment* 19.3, pp. 93–97.

Shi, Lee, Yuhang Zhao, Ricardo Gonzalez Penuela, et al. (2020). "Molder: an accessible design tool for tactile maps". In: *Proceedings of the 2020 CHI conference on human factors in computing systems*, pp. 1–14.

Shi, Shaohuai, Qiang Wang, et al. (2016). "Benchmarking state-of-the-art deep learning software tools". In: *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*. IEEE, pp. 99–104.

Shi, Yuhang Zhao, and Shiri Azenkot (2017). "Markit and Talkit: a low-barrier toolkit to augment 3D printed models with audio annotations". In: *Proceedings of the 30th annual acm symposium on user interface software and technology*, pp. 493–506.

Shih, Lee, and Ku (2016). "A vision-based fingertip-writing character recognition system". In: *Journal of Computer and Communications* 4.4, pp. 160–168.

Shilkrot, Roy et al. (2015). "FingerReader: a wearable device to explore printed text on the go". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 2363–2372.

Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556*.

Son, Young-Jun et al. (2016). "Depth-based fingertip detection for human-projector interaction on tabletop surfaces". In: *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. IEEE, pp. 1–4.

Sun, PanJun (2020). "Security and privacy protection in cloud computing: Discussions and challenges". In: *Journal of Network and Computer Applications* 160, p. 102642.

Symmons, Mark and Barry Richardson (2000). "Raised line drawings are spontaneously explored with a single finger". In: *Perception* 29.5, pp. 621–626.

Thévin, Lauren et al. (2019). "Creating accessible interactive audio-tactile drawings using spatial augmented reality". In: *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, pp. 17–28.

Uijlings et al. (2013). "Selective search for object recognition". In: *International journal of computer vision* 104.2, pp. 154–171.

Vázquez, Marynel and Aaron Steinfeld (2012). "Helping visually impaired users properly aim a camera". In: *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility*, pp. 95–102.

Venkatnarayan, Raghav H and Muhammad Shahzad (2018). "Gesture recognition using ambient light". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.1, pp. 1–28.

Verdadero, Marvin S, Celeste O Martinez-Ojeda, and Jennifer C Dela Cruz (2018). "Hand gesture recognition system as an alternative interface for remote controlled home appliances". In: *2018 IEEE 10th international conference on humanoid, nanotechnology, information technology, communication and control, environment and management (HNICEM)*. IEEE, pp. 1–5.

Virtanen, Pauli et al. (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17, pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

Wang, Wei, Yiyang Hu, et al. (2020). "A new image classification approach via improved MobileNet models with local receptive field expansion in shallow layers". In: *Computational Intelligence and Neuroscience* 2020.

Wang and Bo Yuan (2014). "Robust fingertip tracking with improved Kalman filter". In: *International Conference on Intelligent Computing*. Springer, pp. 619–629.

Wijayawardana, GGAS (2021). "Hand Gesture Pattern Recognition System". PhD thesis.

Wijntjes, Maarten WA et al. (2008a). "Look what I have felt: Unidentified haptic line drawings are identified after sketching". In: *Acta psychologica* 128.2, pp. 255–263.

— (2008b). "The influence of picture size on recognition and exploratory behaviour in raised-line drawings". In: *Perception* 37.4, pp. 602–614.

Wong, Michael, Vishi Gnanakumaran, and Daniel Goldreich (2011). "Tactile spatial acuity enhancement in blindness: evidence for experience-dependent mechanisms". In: *Journal of Neuroscience* 31.19, pp. 7028–7037.

Wu and Wenxiong Kang (2016). "Robust fingertip detection in a complex environment". In: *IEEE Transactions on Multimedia* 18.6, pp. 978–987.

Wu, Wenbin, Chenyang Li, et al. (2017). "Yolse: Egocentric fingertip detection from single rgb images". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 623–630.

Yang, Duan-Duan, Lian-Wen Jin, and Jun-Xun Yin (2005). "An effective robust fingertip detection method for finger writing character recognition system". In: *2005 International Conference on Machine Learning and Cybernetics*. Vol. 8. IEEE, pp. 4991–4996.

You, Yang et al. (2018). "Imagenet training in minutes". In: *Proceedings of the 47th International Conference on Parallel Processing*, pp. 1–10.

Zaman, Mubashira et al. (2016). "Hand gesture recognition using color markers". In: *International Conference on Hybrid Intelligent Systems*. Springer, pp. 1–10.

Zebehazy, Kim T and Adam P Wilton (2014a). "Charting Success: The experience of teachers of students with visual impairments in promoting student use of graphics". In: *Journal of Visual Impairment & Blindness* 108.4, pp. 263–274.

— (2014b). "Quality, importance, and instruction: The perspectives of teachers of students with visual impairments on graphics use by students". In: *Journal of Visual Impairment & Blindness* 108.1, pp. 5–16.

— (2014c). "Straight from the source: Perceptions of students with visual impairments about graphic use". In: *Journal of Visual Impairment & Blindness* 108.4, pp. 275–286.

Zeinullin, Maralbek and Marion Hersh (2022). "Tactile Audio Responsive Intelligent System". In: *IEEE Access* 10, pp. 122074–122091.

Zhang, Valentin Bazarevsky, et al. (2020). "Mediapipe hands: On-device real-time hand tracking". In: *arXiv preprint arXiv:2006.10214*.

Zhang, Fan, Yue Liu, Chunyu Zou, et al. (2018). "Hand gesture recognition based on HOG-LBP feature". In: *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. IEEE, pp. 1–6.

Zhang, Xiaohui, Xinhua Liu, Thompson Sarkodie-Gyan, et al. (2021). "Development of a character CAPTCHA recognition system for the visually impaired community using deep learning". In: *Machine Vision and Applications* 32.1, pp. 1–19.

Zhao, Bardot, Sandra Kaixing, et al. (2021). "Tactile fixations: A behavioral marker on how people with visual impairments explore raised-line graphics". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–12.

Zhao, Yuhang, Shaomei Wu, et al. (2018). “A face recognition application for people with visual impairments: Understanding use beyond the lab”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–14.

Appendix A

Technical Background

The purpose of this section is to give a basic overview of key concepts in the fields of Artificial Intelligence (AI), Computer Vision (CV) and Deep Learning (DL).

Artificial Intelligence

Artificial Intelligence (AI) is the ability of machines and other computer-based systems to execute tasks that typically require human intelligence. The ultimate goal of AI is to mimic human abilities without being explicitly programmed. Machine learning, as its name implies, is an AI subdivision that trains machines to make predictions or take actions based on data inputs. The process of machine learning usually starts with data observation wherein the greater quantity of useful data that is provided to the machine, the more meaningful patterns and conclusions that it can draw. CV is a sub-field of AI that focuses on developing algorithms and systems that are capable of interpreting and understanding visual data. This includes tasks such as image recognition, object detection, and scene understanding. Sometimes biologically inspired neural networks are used by computers to analyse the data. These artificial neural networks work by processing information through layers of interconnected "neurons" which use mathematical operations to learn from input data and make predictions or decisions. This technique is called DL and with the recent increase in computing power, this research area has become quite popular. The combination of CV and DL has led to significant advances in the field and has opened up new applications and possibilities. For example, DL algorithms can be used to analyse large amounts of visual data in real time, enabling applications such as surveillance, traffic management, and augmented reality. Figure 1 illustrates the relationship between the mentioned fields and the research area. In the next section, the fundamentals of CV and DL will be presented.

Computer Vision

CV is a field of AI which enables machines to acquire information from digital photos and videos. CV algorithms range from simple magnification algorithms to complex ones that encapsulate machine and deep learning techniques. Larry Roberts, at that time an MIT Ph.D. student,

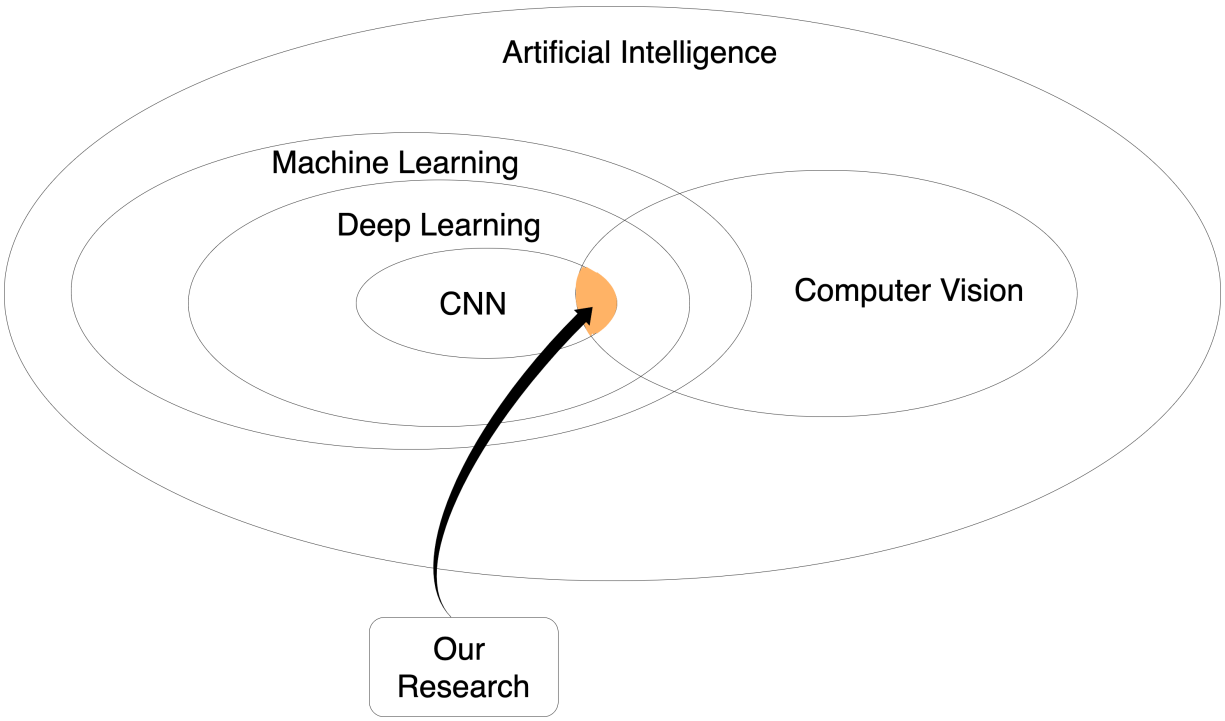


Figure 1: Artificial Intelligence and its fields

laid the foundation of the CV back in the 1960s (Roberts, 1963). He began by studying the machine perception of 3D figures and now this area has advanced to the point where it can be applied to almost any scientific field including the field of AT. CV based systems are widely used to assist VIP in their daily life activities (Budrionis et al., 2022; Zhang, Liu, Sarkodie-Gyan, et al., 2021; Chen, Liu, et al., 2021; Manoharan et al., 2019; Zhao, Wu, et al., 2018). Some of the listed systems rely on the DL algorithms presented below.

Deep Learning

Overview

DL is an area of machine learning which uses artificial neural networks to process data. The concept of artificial neural networks was inspired by the architecture and function of the human brain neuron (Figure 2). Deep Neural Network (DNN) use multiple layers (one input, one or more hidden and one output layers) to enable learning (Figure 3). The first layer deals with raw input data and passes it to the hidden layer, where nonlinear transformations are applied to capture the relations between input features and then sent to the next layer. This process repeats until the generated result reaches the output layer. The more layers a neural network has, the more complex relations that it will detect. Finally, modeled nonlinear relationships are used to make decisions. CNN is a type of DL model that is commonly used in image recognition and processing tasks. The main advantage of CNN is that they are able to learn spatial hierarchies of features, allowing them to effectively handle images and other data with spatial structure.

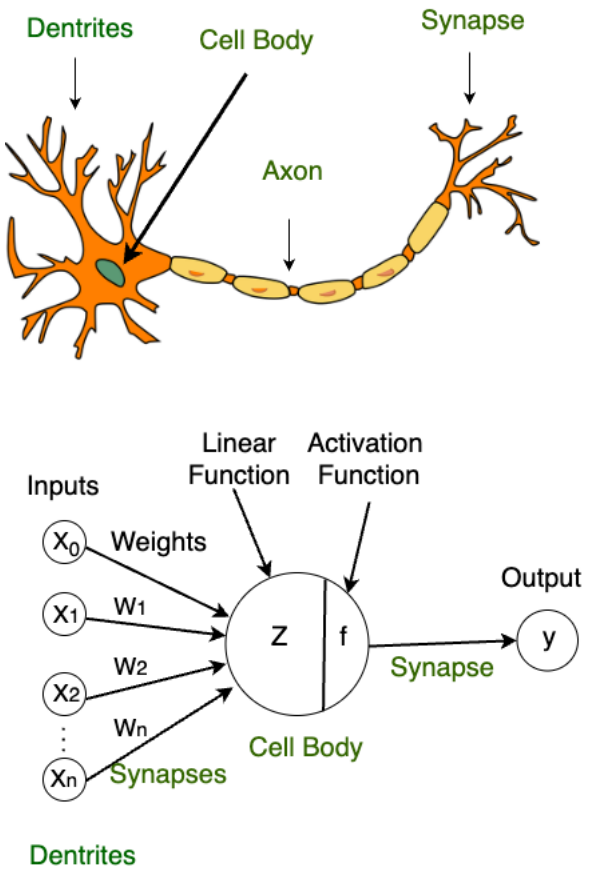


Figure 2: Biological and Artificial neural networks

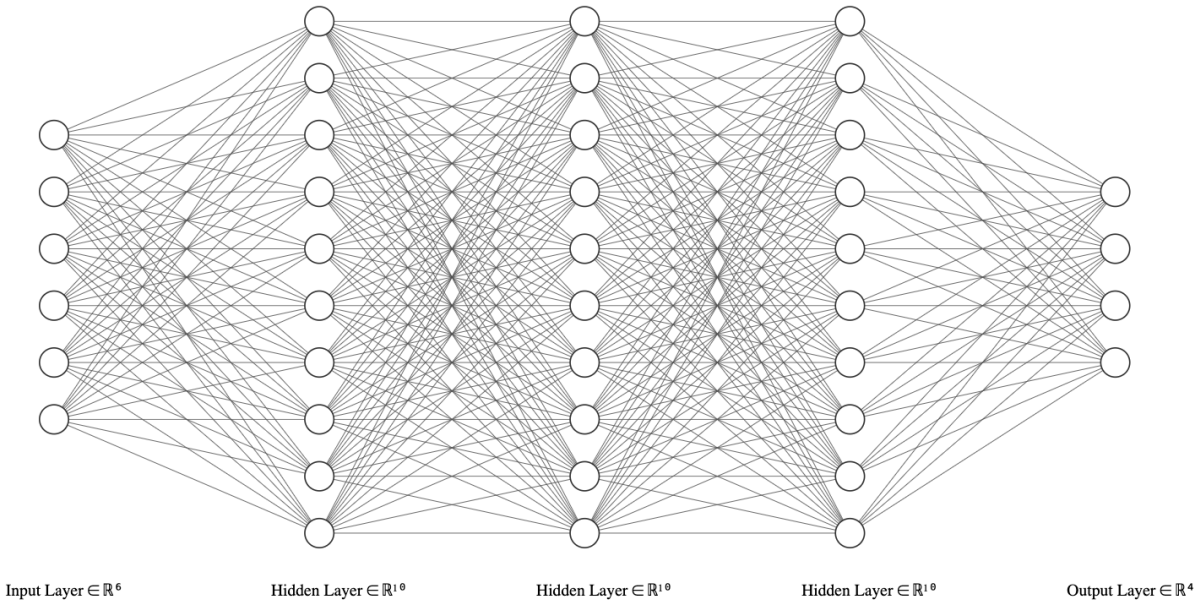


Figure 3: Deep Neural Network with three hidden layers

Convolutional Neural Networks

CNN is a type of neural network architecture that uses convolutional layers to learn spatial hierarchies of features from input data in an automatic and adaptive manner. The term "CNN" or ConvNets was first introduced by LeCun et al. (1989) when he was a postdoctoral student at the University of Toronto. His work was built on the architecture of a Neocognitron multilayered artificial neural network developed by Fukushima and Miyake (1982). An early application of CNN was the recognition of handwritten digits in postal and banking services. Due to the low computational capacities and lack of data at that time, ConvNets could not show their full potential. Only two decades later, Krizhevsky, Sutskever, and Hinton (2012) designed the CNN called AlexNet which created a real breakthrough in the field.

Like other DNN, ConvNets have input, output and hidden layers. The main distinction of this network type is the nature of its hidden part- convolutional layers. As its name suggests, inside these layers, input data is convolved with a **filter** (set of weights that represent a particular feature of the image). The results of these mathematical operations are then summed to produce a single output value. This process is called convolution.

The resultant two-dimensional array of the convolution process is called a **feature map** and it encodes the presence of specific features in the input image. The number of the produced feature maps is equal to the number of filters used. **Activation function** is then applied to each of the feature maps to decide whether it should be activated or not. This function introduces non-linearity into the network, allowing it to learn complex, non-linear relationships between input and output. There are many different activation functions that can be used in DL, including the *sigmoid, tanh and ReLU* functions. Each of these functions has its own characteristics, and choosing the right activation function for a particular network can have a significant impact on its performance. ReLU is often used in CNNs because of its computational efficiency and ability to handle sparse data, sigmoid is used when you want the output to be between 0 and 1, and tanh is used when you want the output to be between -1 and 1.

This process can be repeated multiple times. After going through a certain number of convolutional and activation layers, a **pooling layer** is introduced to reduce the dimensions of the feature maps. The purpose of a pooling layer is to down sample the input, reducing its dimensions and allowing the network to focus on the most important features. There are several different types of pooling layers, but the max pooling layer is the most common. In a max pooling layer, the input is divided into a set of non-overlapping regions, and for each region, the largest value is selected and propagated to the output. This has the effect of retaining only the most important features in the input and discarding the rest. Pooling layers have several benefits: (1) they reduce the computational complexity of the network by reducing the dimensions of the layer, thus allowing it to process inputs more efficiently; (2) they also make the network more resilient to small translations and deformations in the input, improving its generalisation ability.

The last convolutional layer is then connected to **fully connected input layer**. This layer

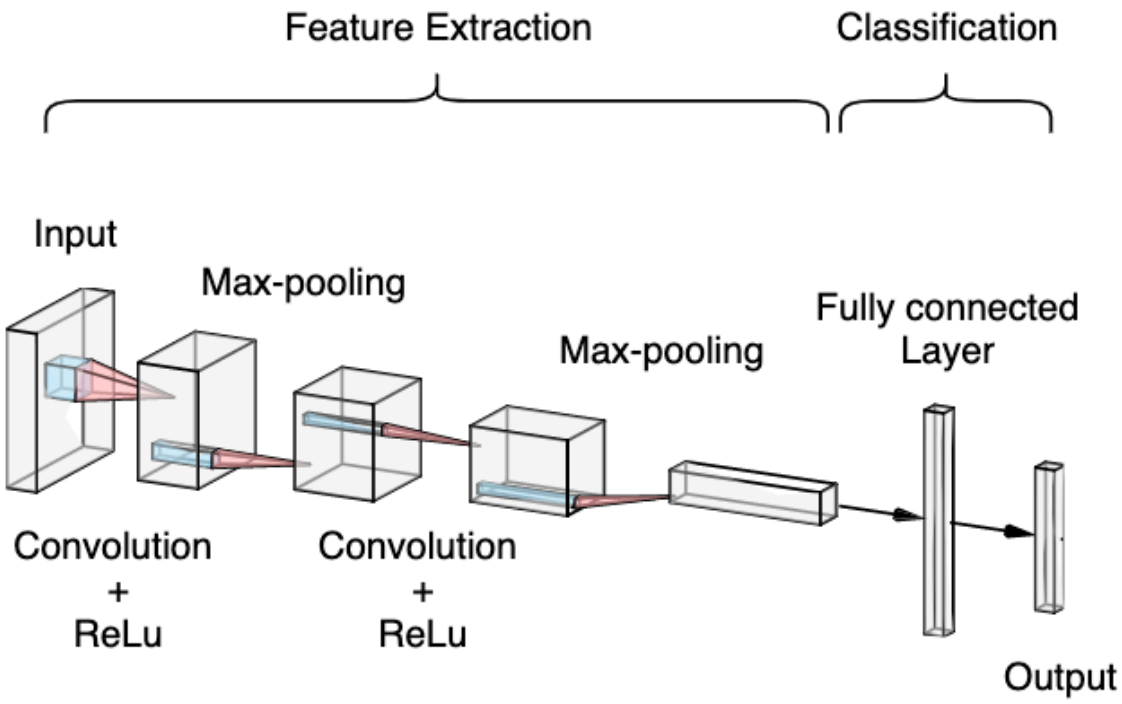


Figure 4: Architecture of CNN

turns the multidimensional data into a single one-dimensional vector and sends the result to the first fully connected layer. Finally, this layer outputs the probability scores for the detected objects. The whole process is illustrated in Figure 4. The first layers of CNN usually detect basic features of the input image (edges, corners, etc.). As we go deeper into the network, more complex features are extracted and the final layers are capable of detecting more semantic information, like faces or whole objects.

ConvNets training requires a large amount of computational power and uses Graphical Processing Unit (GPU) for this process. In general, the main purpose of the training is to calculate proper parameters (weights) for each filter across all layers. Labelled images are the main source of data for CV tasks. To start, networks initialise weights as random numbers. Each image then moves through all layers (forward propagation) before the output is compared to the correct labels. The difference between calculated and ground truth (manually labelled) labels is called a loss. Minimising the loss is the main focus during the CNN model training. After each forward propagation step, a tiny adjustment is made to the weights. The process which is called **backpropagation** helps the network to correct the weights in the right direction. This process based on the idea of propagating errors backwards through the network to update the weights of the connections between neurons, in order to improve the accuracy of the network's output (Rumelhart, Hinton, and Williams, 1986).

In more technical terms, by recursively applying the chain rule to each of the layers of a

network, backpropagation calculates the gradient of a loss function as a function of weights. This involves first computing the error between the predicted and actual outputs of the network, and then propagating this error backwards through the layers of the network. Then, by multiplying it by the derivative of each layer's activation function, and using it to update the weights of the connections between neurons. Multiple iterations are repeated until a set of weights is found that minimises the loss function. Ideally, the loss function should decrease over epochs (sequence of the entire dataset processed) completed. This process can take several hours to weeks, depending on the hardware specifications and the number of images in the dataset. After the training is completed, a test dataset is used to evaluate the model performance. Usually, images that have not been used during the training are used for evaluation. There are two primary CNN object detector types: two-step and one-step detection-based algorithms.

Two-step detectors

Region-based Convolutional Neural Network (R-CNN) family algorithms divide the detection problem into two steps: (i) propose regions (ii) classify the objects within these regions. This approach tends to show a higher accuracy but suffers from low speed. R-CNN method was first introduced by Girshick et al. (2014). The authors used a selective search algorithm created by Uijlings et al. (2013) for the region proposals and the CNN model for the classification. An improved version called Fast R-CNN was then released (Girshick, 2015). This version showed both higher accuracy and speed but similar to its predecessor, it utilised an external region proposal algorithm. Ren et al. (2016) presented a faster R-CNN the same year. The main difference in their approach was the implementation of Region Proposal Network (RPN) instead of a selective search algorithm. RPN is a fully convolutional network that can predict bounding boxes and probability scores for objects simultaneously. This technique showed much faster detection speeds (7 fps for Pascal VOC 2007 testing) but this performance is still very behind the one-step detectors.

One-step detectors

Single-shot Detector (SSD) family, as the name implies, tackles object detection in one step (Liu et al., 2016). SSDs are based on a VGG-16 network (Simonyan and Zisserman, 2014) which serves as a feature map extractor. This network is then connected to the series of 1x1 and 3x3 convolutional layers with different depths. These layers, in their turn, are responsible for object detection. The architecture of the SSD network is illustrated in figure 5. Compared to R-CNNs, SSDs achieve a faster detection rate while maintaining the same level of accuracy (Table 1).

You Only Look Once (YOLO) is another good example of a detector that approaches object detection as a single regression (step) problem. The first version of the YOLO algorithm was proposed by Redmon, Divvala, et al. (2016). The key difference between SSD and YOLO

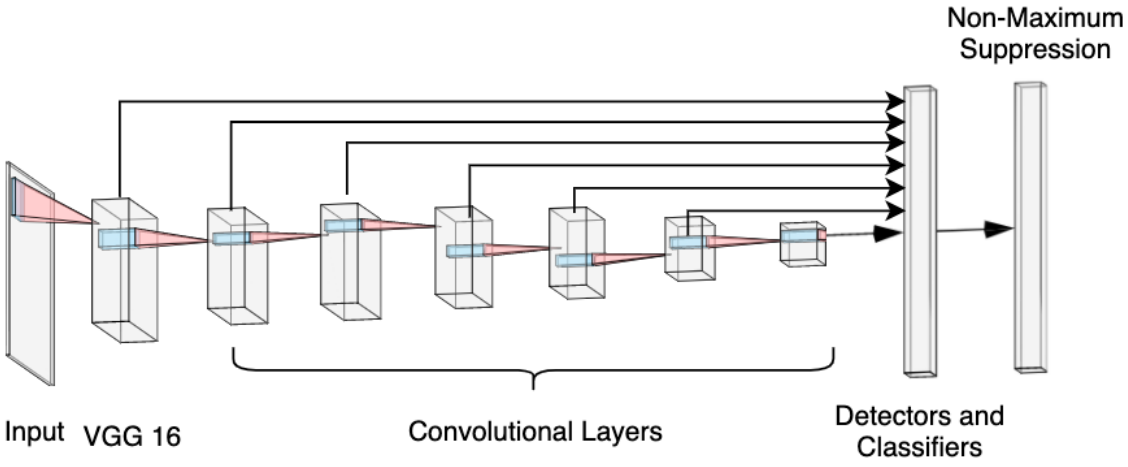


Figure 5: Structure of Single-shot Detector (SSD) network

Method	mAP	FPS	Input image resolution
Faster RCNN (VGG2016)	73.2	7	1000 x 600
SSD300	74.3	46	300 x 300
SSD512	76.8	19	512 x 512

Note. Adapted from Liu et al. (2016)

Table 1: Performance on Pascal VOC2007 test

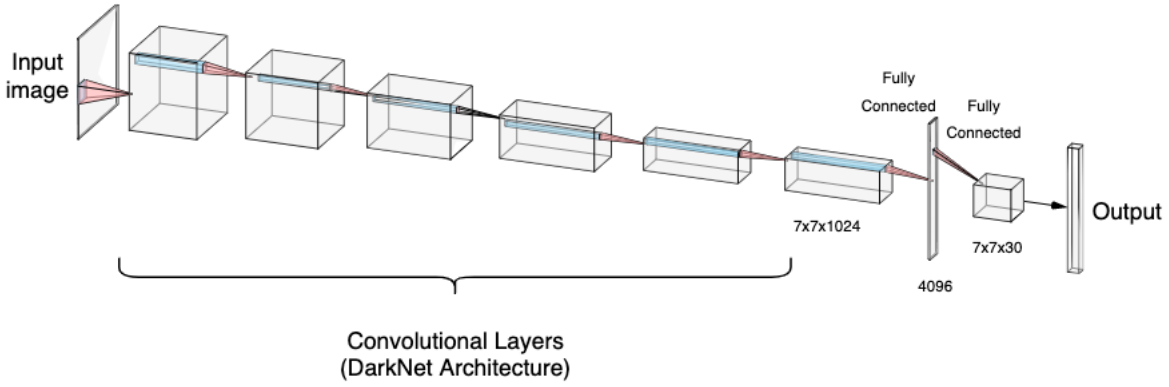


Figure 6: Architecture of YOLO network

architectures is that the latter utilises two fully connected layers instead of convolutional ones to regress the bounding boxes. The object detection process of the YOLO algorithm can be described as follows. First, it divides the input image into a grid of cells, with each cell responsible for predicting a set of bounding boxes. Then, for each cell, YOLO uses a CNN to predict the bounding boxes and their corresponding class probabilities. These predictions are combined across the grid to generate the set of bounding boxes and class probabilities for the image. Once the predictions have been made, YOLO utilises non-maximum suppression to remove overlapping boxes and select the most likely bounding boxes for each object. This helps to improve the accuracy of the predictions and reduce false positives.

The main limitation of the YOLO algorithm is that each cell can detect only one object i.e. if multiple objects fall in the same cell region, only the one with the highest score will be detected. YOLO network architecture is presented in figure 6.

The second version called YOLO9000 was released the next year (Redmon and Farhadi, 2017). The authors improved the network performance by modifying the architecture, adding anchor boxes and introducing other minor optimisations. CNN that has 19 convolutional and 4 max-pooling layers and is called Darknet-19 was used as a backbone for the YOLO9000. To transcend the limitation of a single prediction per grid cell, fully-connected layers were replaced with anchor boxes. Anchors are sets of predefined boxes with a certain width and height. With this feature enabled, the network tries to forecast the object's bounding box as an offset to the anchor, instead of predicting it arbitrarily. The offset can be filtered, so the predictions are adjusted around the predefined shapes. Assessed on the basis of YOLO, this version performs better on the PASCAL VOC 2007 Dataset, showing both better accuracy and higher detection speed (ibid.).

The authors later released an updated version called YOLOv3 (Redmon and Farhadi, 2018). The proposed algorithm was the fastest object detector at that time, tiny YOLOv3 (lighter version), was capable of detecting objects at the rate of 171 fps. This compact version of YOLOv3

was used for the fingertip detection process in my work. A thorough description and analysis of the YOLOv3 is presented in Section 3.2.1.

Deep Learning Frameworks

Like any programming task, DL model development requires a high-level programming interface to start with. This is a user-friendly framework that will enable a programmer to develop a model. State-of-the-art DL frameworks provide a convenient way to build models by facilitating the utilisation of various neural network architectures through the most popular programming languages. Each framework has its unique characteristics and the researchers have to select one which best fits their needs. In this section, the most widely used frameworks will be described and their key features will be compared in a summary table.

TensorFlow (Abadi et al., 2016) is an open-source machine learning framework that utilises data flow graphs for high-performance numerical calculations. This powerful tool was developed by the Google Brain team and its first release was in 2015. The updated version called TensorFlow 2.0 was released in 2019 (this version was used in my research). The main advantage of this framework is an easy deployment on various platforms (desktop, server and edge device) utilising single or multiple Central Processing Unit (CPU), GPU and Tensor Processing Unit (TPU). In addition, this library has an excellent visualisation tool (TensorBoard), well-written documentation and a great selection of publicly available ready-to-use pre-trained models.

Darknet (Redmon, 2013–2016) is another open-source high-performance framework that became well-known after the YOLO algorithm (Redmon, Divvala, et al., 2016) release. The framework was written in C and CUDA[2]. The tiny-YOLOv3 model used in my research was trained using this tool. The main advantage of this framework is that it can be easily configured for GPU training.

PyTorch (Paszke et al., 2019) is a relatively new DL framework developed by Facebook and written in Python. As a result, it has a cleaner interface and Python developers will find it easier to use. Additionally, this framework is compatible with NumPy (Harris et al., 2020), one of the most popular scientific computing libraries.

Keras (Chollet et al., 2015) is a user-friendly and easy-to-use high-level DL framework. This makes it extremely popular among those who just started exploring the DL field. Fast experimentation with DNN is possible with this framework. It is worth mentioning that Keras uses TensorFlow as its default backend computational engine. Thus, it is easy to call the classes and functions of Tensorflow without adding any additional code.

Caffe (Jia et al., 2014) is an extremely fast DL framework developed by Berkeley Vision and Learning Centre (BVLC). According to their website[3], this framework is capable of processing

[2]https://developer.nvidia.com/cuda-zone
[3]https://caffe.berkeleyvision.org/

over 60 million images with a single NVIDIA K40 GPU per day. Thus, Caffe models can be easily deployed on mobile or edge devices. Another advantage of this framework is a Caffe model zoo- an open-source repository of pre-trained models.

To conclude, in this section some of the most popular DL frameworks were discussed. Their general information, together with their advantages and disadvantages, is summarised in Table 2. This list is by no means exhaustive and other widely used DL frameworks include: Theano (Al-Rfou et al., 2016), MXNet (Chen, Li, et al., 2015), ONNX (Bai, Lu, and Zhang, 2019), CNTK (Seide and Agarwal, 2016) and others. In the context of the research, TensorFlow 2 and Darknet frameworks were used for the object detection models training. TensorFlow 2 was selected due to its object detection API that makes the training process very easy and intuitive. Thus, it is very convenient for experimenting. Whereas Darknet was the most suitable for training models which could be easily deployed on mobile devices.

DL Framework	**Year Released**	**Interface**	**Pros**	**Cons**
TensorFlow	2015	Python	Scalable, pre-trained models available, supported by Google	Comparatively slow, frequent updates and tricky to configure with GPU
Darknet	2013	C, C++, Python	Easy to install, pre-trained models available	Models tend to be less accurate
PyTorch	2017	Python	Dynamically updated graph, good for experimenting and research, supported by Facebook	Lack of visualisation tools, small developer community
Keras	2015	Python	Easy to use, uses other frameworks as its backend, pre-trained models available	Slower than its backend, difficult to debug
Caffe	2013	C++, Python	Extemely fast, pre-trained models available	Not scalable

Table 2: Most used DL Frameworks

Summary

CV is a field of computer science that focuses on enabling computers to interpret and understand visual data. It involves developing algorithms and models that can automatically analyse and understand visual information in order to perform tasks such as object recognition, image segmentation and scene understanding.

DL is a type of machine learning that involves using artificial neural networks to learn complex patterns in data. These neural networks are made up of many layers of interconnected nodes, which can be trained to recognise and classify different objects and features in images and videos.

In the context of my research, CV and DL techniques were used to develop an algorithm that can detect and track fingertips in real-time and facilitate user interaction with the developed app.

Appendix B

Tactile Graphics Descriptions and Questions

Space Shuttle Description

This is an image of a space shuttle. It is a partly reusable space vehicle in which people travel into space and back again. It is also used for carrying a satellite or other equipment into orbit. It consists of an external fuel tank, rocket boosters, orbiter and its main engines.

External fuel tank: It carries the fuel for the Orbiter main engines. Also it connects the orbiter with the rocket boosters. It jettisons after the launch and not reused.

Rocket boosters: Rockets used to launch and accelerate a space shuttle during liftoff. After burnout, they jettison and parachute into the ground where they examined, refurbished, and reused.

Orbiter: It is the spaceplane component of the shuttle which goes to the orbit.

Main engines: They aid the orbiter to reach the orbit after the rocket boosters are jettisoned.

Space Shuttle Questions

1. How many rocket boosters does the shuttle have?
2. Which component of the space shuttle is not reused:
   (a) Orbiter
   (b) Rocket boosters
   (c) External tank
   (d) Main engines
3. Which component of the space shuttle reaches the orbit together with the orbiter?
4. **(Memory question)** What are the four components of the space shuttle?

Tactile Graphics Descriptions and Questions

Frog Life Cycle Description

This is an image of the frog life cycle. There are six stages presented in a clockwise direction: eggs, tadpole, tadpole with two legs, tadpole with four legs, froglet, and adult frog.

Eggs: A frog begins life as a fertilized egg. An adult frog lays hundreds of eggs at one time.

Tadpole: The tadpole has an oval-shaped body and a long tail.

Tadpole with two legs: At this stage, the tadpole develops back legs while still retaining its tail.

Tadpole with four legs: At this stage, the tadpole has both front and back legs along with its tail.

Froglet: The froglet possesses pairs of front and back legs, a larger body, and a shortened tail.

Adult frog: At this stage, the tail disappears completely. Adult frogs lay eggs to begin the life cycle anew.

Frog Life Cycle Questions

1. How many stages are there in the frog life cycle?
2. At what stage does the tail disappear completely?
   (a) Tadpole
   (b) Adult frog
   (c) Tadpole with four legs
   (d) Froglet
3. Which legs grow first?
4. **(Memory question)** Name all stages of the frog life cycle without using the tactile graphic.

Map of Australia Description

This is the map of Australia. It consists of six states and two territories.

Western Australia: The capital city is Perth.

Northern Territory: The capital city is Darwin.

Queensland: The capital city is Brisbane.

South Australia: The capital city is Adelaide.

New South Wales: The capital city is Sydney. The Australian Capital Territory is located in this state as well.

Victoria: The capital city is Melbourne.
Tasmania: The capital city is Hobart.

Map of Australia Questions

1. Which state or territory is surrounded by Western Australia, Queensland, and South Australia?

2. What is the capital of New South Wales?

   (a) Sydney

   (b) Brisbane

   (c) Perth

   (d) Adelaide

3. Which state is a separate island?

4. **(Memory question)** Without using the tactile graphic, how many states and territories are there in Australia?

Map of Kyrgyzstan Description

This is the map of Kyrgyzstan. It consists of seven regions.
Talas Region: The capital is Talas.
Chui Region: The capital is Bishkek.
Issyk-Kul Region: The capital is Karakol.
Jalal-Abad Region: The capital is Jalal-Abad.
Naryn Region: The capital is Naryn.
Osh Region: The capital is Osh.
Batken Region: The capital is Batken.

Map of Kyrgyzstan Questions

1. What is the capital of the Issyk-Kul Region?

2. In which region is the capital of the country located?

   (a) Talas

   (b) Chui

   (c) Issyk-Kul

   (d) Osh

3. Which region is located between the Talas and Osh regions?

4. **(Memory question)** Without using the tactile graphic, how many regions are there in Kyrgyzstan?

Histogram Description

This histogram represents the number of books read over six months.

Months: Below are the months and the corresponding number of books read: **June:** Number of books read in June is 3.

May: Number of books read in May is 5.

April: Number of books read in April is 2.

March: Number of books read in March is 0.

February: Number of books read in February is 1.

January: Number of books read in January is 1.

Number of Books Read: The histogram shows the count of books read above each month.

Histogram Questions

1. During which month were the most books read?

2. During which months were no books read?

   (a) January

   (b) February

   (c) March

   (d) April

3. During which two months was the same number of books read?

4. **(Memory question)** Without using the tactile graphic, how many months' statistics are presented in the histogram?

Percentage Graph Description

This graph represents the library occupancy throughout the week.

Below are the days of the week and the corresponding library occupancy percentages:

Monday: Library occupancy on Monday is 100

Tuesday: Library occupancy on Tuesday is 60

Wednesday: Library occupancy on Wednesday is 20

Thursday: Library occupancy on Thursday is 80

Friday: Library occupancy on Friday is 40 **Saturday:** Library occupancy on Saturday is 0
Sunday: Library occupancy on Sunday is 40

Percentage Graph Questions

1. On what day is the library full?
2. What is the library occupancy on Wednesday?
   (a) 80%
   (b) 60%
   (c) 40%
   (d) 20%
3. On which two days is the library occupancy the same?
4. **(Memory question)** Without using the tactile graphic, on what day is the library empty?

Interview Questions

Personal information for statistical purposes

1. Please select the option that best describes you
   Secondary school student
   University student
   College student or a vocational trainee
   A person who has left education in the last five years
   Other (Please specify)
2. What is your age?
3. Which gender do you identify with?
   Male
   Female
   Other
4. What country do you live in?
5. Which of the following best describes your vision?
   Totally blind with no awareness of light
   Blind and able to distinguish light and dark
   Able to see shapes, but unable to distinguish detail or to read print
   Able to read a large print text
   Other (Please specify)

6. Which of the following statements best describes you?
   I have been blind or partially sighted since birth
   I lost my sight between 0 and 3 years
   I lost my sight between 3 and 11 years
   I lost my sight between 11 and 35 years
   I lost my sight after the age of 35
7. Which of the following best describes the school you are attending now or attended most recently? (Check all that apply)
   School for the blind
   A mainstream school with resource room
   The mainstream school (full-time classroom)
   Other (Please specify)
8. Can you read Braille?
   Yes
   No
   I understand some Braille but I am not fluent
9. Have you used tactile graphics?
   Yes, in education
   Yes, in other applications but not education
   No

Please, proceed to section B if your answer to the previous question was "YES".
If your answer was "NO", please answer the next two questions and you can skip section B.

10. Would you like to use tactile graphics?
    Yes
    No (please specify the reason)
    Unsure (please specify the reason)
11. For what reasons have you not used them?
    Not provided by teachers
    No training was provided on how to use them
    Other (please specify)

Your experience with tactile graphics

12. What subjects have you used tactile graphics in? (Check all that apply)
    STEM (Science, technology, engineering and mathematics)
    Geography
    History
    Orientation and Mobility classes

Art and music

Domestic science and woodwork

Other (please specify)

12. How were these tactile graphics usually labelled?

Braille text

Large printed text

Audio descriptions

Not labelled

Other (please specify)

14. How often do you use tactile graphics in education?

Most of the time (3-4 times a week or more)

Sometimes (1-2 times a week)

Rarely (a few times a month or less)

Almost never

15. Please select the option that best applies to you

Tactile graphics are sufficient on their own without Braille and text descriptions and help me understand the concept better

Tactile graphics complement Braille and text descriptions and help me understand the concept better

Tactile graphics do not help me understand the concept better

16. Is it easier to keep up when tactile graphics are used in class?

Yes, they make a big difference

Yes, they help a bit

No, they make no difference

No, I can keep up without them

17. Indicate all of the following that would help you to use tactile graphics more effectively

Training in using tactile graphics

Being shown how to use/oriented to each tactile graphics

Audio descriptions provided with tactile graphics

Training in using tactile graphics with audio descriptions

Other (please specify)

18. Please give three examples of how you have used tactile graphics to support your learning with comments on how useful you found them

19. Please comment on any features of tactile graphics you have used that makes them easy or difficult to read and any strategies you have used to read new tactile graphics.

20. Do you have any suggestions on how tactile graphics could be improved?

Your experience with assistive technology phone applications

21. Which mobile phone do you use?

    Android smartphone

    IOS smartphone

    Another smartphone

    Phone without a camera

    I do not use a mobile phone

22. If you ever used an app which requires you to use a phone camera, which of the following apps have you used? (Check all that apply)

    Camera (for taking pictures)

    Currency reader

    Color identifier

    Photo Identifier (e.g. VizWiz)

    Light detector

    Barcode reader

    Object identifier (e.g. TapTapSee)

    Other (Please specify)

23. How often do you use an app which requires you to use a camera?

    Everyday

    Once or twice a week

    Once a month

    Almost never

24. Do you want applications to give you feedback to properly aim the camera?

    No, I can aim the camera without feedback

    Yes, I would use camera apps more frequently if there was feedback on how to aim the camera

    Yes, I would consider using camera apps if feedback to aim the camera was available

25. How the feedback should be conveyed to you while aiming the camera?

    Vibration

    Voice information

    Tones

    A combination of the above

    Other, please specify

Appendix C

TAURIS Mobile Application Source Code

MainActivity.java

```
package com.example.YoloDetectionFiveFingers;
import android.content.res.AssetManager;
import android.content.Context;
import android.os.Build;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import android.os.StrictMode;
import android.os.VibrationEffect;
import android.os.Vibrator;
import android.speech.tts.TextToSpeech;
import android.util.Log;
import android.view.SurfaceView;
import android.view.WindowManager;
import android.widget.Toast;
import org.opencv.android.BaseLoaderCallback;
import org.opencv.android.CameraBridgeViewBase;
import org.opencv.android.JavaCameraView;
import org.opencv.android.OpenCVLoader;
import org.opencv.aruco.Aruco;
import org.opencv.aruco.Dictionary;
import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.core.MatOfFloat;
import org.opencv.core.MatOfInt;
import org.opencv.core.MatOfRect;
import org.opencv.core.Point;
import org.opencv.core.Rect;
import org.opencv.core.Scalar;
import org.opencv.core.Size;
import org.opencv.dnn.Net;
import org.opencv.imgproc.Imgproc;
import org.opencv.dnn.Dnn;
```

```
import org.opencv.utils.Converters;
import org.opencv.android.FpsMeter;
import org.opencv.objdetect.QRCodeDetector;
import java.io.BufferedInputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.Locale;
import java.lang.String;
import static java.lang.System.out;


public class MainActivity extends AppCompatActivity implements
    CameraBridgeViewBase.CvCameraViewListener2 {
    private static final String TAG = "OCVSample::Activity";
    CameraBridgeViewBase cameraBridgeViewBase;
    BaseLoaderCallback baseLoaderCallback;
    public TextToSpeech tts1;
    String newResult; //This variable needed to compare old and new text
        which is to be converted to speech
    boolean fetched = false;
    boolean QRdetected = false;
    boolean datafetched = false;
    boolean tell_description = false;
    int step = 40;
    int counter = 0;
    int vibro1 = 0;
    int vibro2 = 0;
    int timer = 0;

    String final_planet = "false";

    FpsMeter fpsMeter = new FpsMeter();

    Net tinyYolo;
    public static String tinyYoloCfg;
    public static String tinyYoloWeights;
    private static String getPath(String file, Context context) {
        AssetManager assetManager = context.getAssets();
        BufferedInputStream inputStream = null;
        try {
            // Read data from assets.
            inputStream = new BufferedInputStream(assetManager.open(file));
```

```
            byte[] data = new byte[inputStream.available()];
            inputStream.read(data);
            inputStream.close();
            // Create copy file in storage.
            File outFile = new File(context.getFilesDir(), file);
            FileOutputStream os = new FileOutputStream(outFile);
            os.write(data);
            os.close();
            // Return a path to file which may be read in common way.
            return outFile.getAbsolutePath();
        } catch (IOException ex) {
            Log.i(TAG, "Failed to upload a file");
        }
        return "";
    }


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        if (android.os.Build.VERSION.SDK_INT > 9)
        {
            StrictMode.ThreadPolicy policy = new
                    StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
        }
        cameraBridgeViewBase = (JavaCameraView)findViewById(R.id.CameraView)
            ;
        cameraBridgeViewBase.setVisibility(SurfaceView.VISIBLE);
        cameraBridgeViewBase.setCvCameraViewListener(this);
        getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON)
            ;
        baseLoaderCallback = new BaseLoaderCallback(this) {
            @Override
            public void onManagerConnected(int status) {
                super.onManagerConnected(status);
                switch(status){
                    case BaseLoaderCallback.SUCCESS:
                        cameraBridgeViewBase.enableView();
                        break;
                    default:
                        super.onManagerConnected(status);
                        break;
                }
            }
```

```
        };

        tts1=new TextToSpeech(getApplicationContext(),
                new TextToSpeech.OnInitListener() {
                    @Override
                    public void onInit(int status) {
                        if(status != TextToSpeech.ERROR){
                            tts1.setLanguage(Locale.getDefault());
                        }
                    }
                });

    }
    public void convertTextToSpeech(String text) {
        if (null == text || "".equals(text)) {
            Log.d(TAG, "Nothing to say");
        }
        tts1.speak(text, TextToSpeech.QUEUE_FLUSH, null, TextToSpeech.
            ACTION_TTS_QUEUE_PROCESSING_COMPLETED);
    }



    //initialize YoloDetector method. Input frame and transformation matrix
        are the parameters.
    public Mat yoloDetector(Mat frame, Mat perspectiveTransformation){
        //convert RGBA to RGB
        Imgproc.cvtColor(frame, frame, Imgproc.COLOR_RGBA2RGB);
        Mat imageBlob = Dnn.blobFromImage(frame, 0.00392, new Size(416,416),
            new Scalar(0, 0, 0),/*swapRB*/false, /*crop*/false);
        tinyYolo.setInput(imageBlob);
        java.util.List<Mat> result = new java.util.ArrayList<Mat>(2);
        List<String> outBlobNames = new java.util.ArrayList<>();
        outBlobNames.add(0, "yolo_16");
        outBlobNames.add(1, "yolo_23");
        tinyYolo.forward(result,outBlobNames);
        float confThreshold = 0.2f;
        List<Integer> clsIds = new ArrayList<>();
        List<Float> confs = new ArrayList<>();
        List<Rect> rects = new ArrayList<>();
        boolean middledetected = false;
        boolean ringdetected = false;
        boolean pinkydetected = false;
        boolean indexdetected = false;
        List<String> cocoNames = Arrays.asList("Thumb", "Index", "Middle", "
            Ring", "Pinky");
```

```
int intConf = 0;
Rect box = new Rect(0,0,0,0);

for (int i = 0; i < result.size(); ++i)
{
    Mat level = result.get(i);
    for (int j = 0; j < level.rows(); ++j)
    {
        Mat row = level.row(j);
        Mat scores = row.colRange(5, level.cols());
        Core.MinMaxLocResult mm = Core.minMaxLoc(scores);
        float confidence = (float)mm.maxVal;
        Point classIdPoint = mm.maxLoc;
        if (confidence > confThreshold)
        {
            int centerX = (int)(row.get(0,0)[0] * frame.cols());
            int centerY = (int)(row.get(0,1)[0] * frame.rows());
            int width   = (int)(row.get(0,2)[0] * frame.cols());
            int height  = (int)(row.get(0,3)[0] * frame.rows());
            int left    = centerX - width  / 2;
            int top     = centerY - height / 2;
            clsIds.add((int)classIdPoint.x);
            confs.add((float)confidence);
            rects.add(new Rect(left, top, width, height));
        }
    }
}

int ArrayLength = confs.size();
if (ArrayLength >= 1) {
    // Apply non-maximum suppression procedure.
    float nmsThresh = 0.2f;
    MatOfFloat confidences = new MatOfFloat(Converters.
        vector_float_to_Mat(confs));
    Rect[] boxesArray = rects.toArray(new Rect[0]);
    MatOfRect boxes = new MatOfRect(boxesArray);
    MatOfInt indices = new MatOfInt();
    Dnn.NMSBoxes(boxes, confidences, confThreshold, nmsThresh,
        indices);
    // Draw result boxes:
    int[] ind = indices.toArray();
    for (int i = 0; i < ind.length; ++i) {
        int idx = ind[i];
        //idGuy is the id of the finger. 0 is thumb, 1 is index and
            so on.
        int idGuy = clsIds.get(idx);
```

```
            float conf = confs.get(idx);

            //if detected finger is index, we need to store its
               confidence value and surrounding box coordinates. We
               will use later to detect the center point
            if(idGuy == 1) {
                indexdetected = true;
                intConf = (int) (conf * 100);
                box = boxesArray[idx];
            }

            //if detected finger is middle, we need to change the
               boolean to true
            else if(idGuy == 2) {
                middledetected = true;
            }

            //if detected finger is ring, we need to change the boolean
               to true
            else if(idGuy == 3) {
                ringdetected = true;
            }

            //if detected finger is pinky, we need to change the boolean
                to true
            else if(idGuy == 4) {
                pinkydetected = true;
            }
        }

        //we proceed only if all fingers are detected. Thumb is not that
            crucial in our case. Also we return the location of index
           finger only
        //***change this part 28/08/2020
        if(indexdetected == true && middledetected == true &&
           ringdetected == true && pinkydetected == true) {
            //step is the dimensions of the square cell in pixels
            int step = 15;

                // ===========  KALMAN FILTER APPLICATION ===========

        // 1. Prepare Measurement
        measurement.put(0, 0, new_centerX);
        measurement.put(1, 0, new_centerY);

        // 2. Predict using Kalman Filter
```

```
Mat prediction = kalmanFilter.predict();
Point predictedPt = new Point(prediction.get(0, 0)[0],
   prediction.get(1, 0)[0]);

// 3. Correct Kalman Filter with Measurement
kalmanFilter.correct(measurement);

    //here use predictedPt (x, y from Kalman filter) for cell
       mapping
    //    instead of new_centerX and new_centerY
    //box top left angle added by the half of the width/height
    double centerX = box.x + box.width/2;
    double centerY = box.y + box.height/2;
    //here we find the desired points coordinates in the warped
       image. dst = H * src. Where H is the transformation
       matrix taken from the function parameter
    double new_centerX = (perspectiveTransformation.get(0,0)[0]*
       centerX + perspectiveTransformation.get(0,1)[0]*centerY
       + perspectiveTransformation.get(0,2)[0])/(
       perspectiveTransformation.get(2,0)[0]*centerX +
       perspectiveTransformation.get(2,1)[0]*centerY +
       perspectiveTransformation.get(2,2)[0]);
    double new_centerY = (perspectiveTransformation.get(1,0)[0]*
       centerX + perspectiveTransformation.get(1,1)[0]*centerY
       + perspectiveTransformation.get(1,2)[0])/(
       perspectiveTransformation.get(2,0)[0]*centerX +
       perspectiveTransformation.get(2,1)[0]*centerY +
       perspectiveTransformation.get(2,2)[0]);
    //normalize the width and height to 900 and 600 respectively
    new_centerX = new_centerX/GetWarpedFrame.maxWidth*
       GetWarpedFrame.newWidth;
    new_centerY = new_centerY/GetWarpedFrame.maxHeight*
       GetWarpedFrame.newHeight;
    //proceed only if the finger position is within the area
       enclosed by markers
    if (new_centerX >= 0 && new_centerX <= 900 && new_centerY >=
        0 && new_centerY <= 600  ) {
        //find the corresponding cell numbers. Image is 60 (900/
           step) x 40 (600/step) = 2400 cell sized.
        int cell_x = (int) Math.ceil(new_centerX / step);
        int cell_y = (int) Math.ceil(new_centerY / step);
        // N is the number of cells in one row
        int N = GetWarpedFrame.newWidth / step;
        //int N = GetWarpedFrame.newHeight / step;

        //cell_number is the unique value of each of the
```

```
    900*600/(15*15) = 2400 cells
int cell_number = N * Math.abs(cell_y - 1) + cell_x;
//int cell_number = N * Math.abs(cell_y - 1) + cell_x;


//access cell element from the excel file using the cell
     number and convert it  to String

//proceed only if the cell value is within the array
    size
if (cell_number >= 0 && cell_number <= 2400) {
     System.out.println("cell number is " + cell_number);
     // catch exception if json file was not downloaded
     try {
          String cell_number_string = String.valueOf(
              cell_number);
          final_planet = fetchData.dict.get(
              cell_number_string);
     } catch(IndexOutOfBoundsException e) {
          //prevent from multiple TTS
          if (datafetched == false) {
               convertTextToSpeech("Please check your
                   internet connection");
               datafetched = true;
          }
          final_planet = "false";
     }
     //Only TTS text when the object location is not
        empty and is changed so one text is not repeated
         many times
     if (!final_planet.equals(newResult) && !"false".
        equals(final_planet)) {
          convertTextToSpeech(final_planet);
          newResult = final_planet; //this is used to
              compare old and new text
          timer = 0;
     }

     else if (final_planet.equals(newResult)) {
          timer = timer + 1;
          Log.d("timer: ", "> " + timer);   //here u ll
              get whole response..... :-)

          if (timer == 6){
               String description = fetchData.dict.get(
                   final_planet);
```

```
                            convertTextToSpeech(description);
                            timer = 0;
                        }
                    }

                    else if ("false".equals(final_planet)) {
                        timer = 0;
                    }


                    //draw the box around the index frame
                    Imgproc.putText(frame, cocoNames.get(1) + " " +
                        intConf + "%", box.tl(), Imgproc.
                        FONT_HERSHEY_COMPLEX, 2, new Scalar(0, 0, 0), 2)
                        ;
                    Imgproc.rectangle(frame, box.tl(), box.br(), new
                        Scalar(255, 0, 0), 2);
                }
            }
        }
    }
    //return the frame with bounding box
    return frame;
}


@Override
public Mat onCameraFrame(CameraBridgeViewBase.CvCameraViewFrame
    inputFrame) {
    Mat frame = inputFrame.rgba();

    if(QRdetected == false){

        QRCodeDetector qrDecoder = new QRCodeDetector();
        String QRtext = qrDecoder.detectAndDecode(frame);
        //new fetchData().execute("https://api.jsonbin.io/b/5
            ea97ac94c87c3359a63bd78");//Uk map
        //https://api.jsonbin.io/b/5ea9bb884c87c3359a63db73
        if (counter % step ==0){
            //convertTextToSpeech("Please scan the QR code first");
            convertTextToSpeech("
                                                        ");

        }
        counter+=1;
```

```
            if(QRtext.contains("http")) {
                Log.d("QRcode text: ", QRtext);
                new fetchData().execute(QRtext);//shapes
                QRdetected = true;
                //convertTextToSpeech("QR code scanned successfully");
                convertTextToSpeech("
                                              ");



            }


        }

        if(QRdetected == true) {

            Mat ids = new Mat();// needed for Aruco
            List<Mat> corners = new ArrayList<>(); // needed for Aruco
            Dictionary dictionary = Aruco.getPredefinedDictionary(Aruco.
                DICT_4X4_250); // needed for Aruco
            Mat perspectiveTransformation = new Mat();
            //Start detecting Aruco markers
            Aruco.detectMarkers(inputFrame.gray(), dictionary, corners, ids)
                ;
            // Initiate vibration
            Vibrator v = (Vibrator) getSystemService(Context.
                VIBRATOR_SERVICE);

            // Initiate descriptions when at least one marker is visible
            // it will initiated only once
            if (ids.size(0) >= 2 && tell_description == false) {
                if(fetchData.dict.get("title") != null && !fetchData.dict.
                    get("title").trim().isEmpty()) {
                    String title = fetchData.dict.get("title");
                    convertTextToSpeech(title);
                    tell_description = true;
                }
            }

            int sum = 0;
            // vibrate when more than one and less than three markers are
                visible
            if (ids.size(0) == 1) {
                //initiate vibro counter
```

```
                vibro1 = vibro1+1;
                //System.out.println("Vibro 1 : "+ vibro1);

                //if markers are not visible on 50 consecutive frames
                   vibrate
                if (vibro1 == 50) {
                    // take care of API versions deprecation
                    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                        v.vibrate(VibrationEffect.createOneShot(100,
                           VibrationEffect.EFFECT_DOUBLE_CLICK));
                        vibro1 = 0;
                    } else {
                        //deprecated in API 26
                        //pattern: 0-start without delay, 50- duration, 50-
                           pause, 50-duration (double vibration)
                        long[] pattern = {0, 50, 50, 50};
                        // -1 = no repeat
                        v.vibrate(pattern, -1);
                        // start over the counter
                        vibro1 = 0;
                    }
                }
            }

            if (ids.size(0) == 2) {

                //initiate vibro2 counter
                //Notify the user that only two markers are visible and it
                   is advised to recalibrate by making at least 3 markers
                   visible
                vibro2 = vibro2+1;
                //System.out.println("Vibro 2 : "+ vibro2);

                //if markers are not visible on 10 consecutive frames
                   vibrate
                if (vibro2 == 10) {
                    // take care of API versions deprecation
                    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                        v.vibrate(VibrationEffect.createOneShot(100,
                           VibrationEffect.EFFECT_DOUBLE_CLICK));
                        vibro2 = 0;
                    } else {
                        //deprecated in API 26
                        //pattern: 0-start without delay, 50- duration (
                           single vibration)
                        long[] pattern = {0, 50};
```

```
                // -1 = no repeat
                v.vibrate(pattern, -1);
                // start over the counter
                vibro2 = 0;
            }
        }

        fpsMeter.measure();
        int size = 2;
        //iterate over each Aruco marker
        for (int i = 0; i < 2; i++) {
            int ID = (int) ids.get(i, 0)[0];
            if (ID < 4) { //sometimes wrong ID numbers are detected.
                 So restricted them to <4
                Mat markerCorners = corners.get(i);
                //call the CornerPoints class
                CornerPoints pointvalues = new CornerPoints(
                   markerCorners, ID, size);
                //call the method within that class
                pointvalues.getMarkerCorners();
            }
        }
        //it looks like Aruco uses old detected coordinate values if
             new ones are not detected
        //get 2D arrays of 4 corner points. Only one corner of each
           marker is detected to get a new frame enclosed by inner
           corners
        int corner_tr[][] = {{CornerPoints.markerfeatures[0][2][1]},
             {CornerPoints.markerfeatures[0][2][2]}}; //bottom left
           corner of TL marker
        int corner_br[][] = {{CornerPoints.markerfeatures[1][3][1]},
             {CornerPoints.markerfeatures[1][3][2]}}; //BR corner of
             TR marker
        int corner_bl[][] = {{CornerPoints.markerfeatures[2][0][1]},
             {CornerPoints.markerfeatures[2][0][2]}}; //Tl corner of
             BR marker
        int corner_tl[][] = {{CornerPoints.markerfeatures[3][1][1]},
             {CornerPoints.markerfeatures[3][1][2]}}; //TR corner of
             BL marker
        GetWarpedFrame finalframe = new GetWarpedFrame(frame,
           corner_tr, corner_br, corner_bl, corner_tl);
        perspectiveTransformation = finalframe.getTransform();
        frame = yoloDetector(frame, perspectiveTransformation);
    }
```

```
//if only 3 markers are detected
if (ids.size(0) == 3) {
    vibro1 = 0;
    vibro2 = 0;
    fpsMeter.measure();

    int size = 3;
    for (int i = 0; i < 3; i++) {
        int ID = (int) ids.get(i, 0)[0];
        sum = ID + sum;
    }
    // If missing ID is 0 (1+2+3=6). 0 is not considered because
        detected markers size is 3
    if (sum == 6) {
        for (int i = 0; i < 3; i++) {
            int ID = (int) ids.get(i, 0)[0];
            Mat markerCorners = corners.get(i);
            CornerPoints pointvalues = new CornerPoints(
               markerCorners, ID, size);
            pointvalues.getMarkerCorners();
        }
        int corner_br[][] = {{CornerPoints.markerfeatures
           [1][3][1]}, {CornerPoints.markerfeatures[1][3][2]}};
            //BR corner of TR marker
        int corner_bl[][] = {{CornerPoints.markerfeatures
           [2][0][1]}, {CornerPoints.markerfeatures[2][0][2]}};
            //Tl corner of BR marker
        int corner_tl[][] = {{CornerPoints.markerfeatures
           [3][1][1]}, {CornerPoints.markerfeatures[3][1][2]}};
            //TR corner of BL marker
        //missing corner is calculated from the remaining 3
           corners. Add x and y values of diagonal corner
           points and subtract the corner points of the
           remaining marker
        int corner_tr[][] = {{corner_tl[0][0] + corner_br[0][0]
           - corner_bl[0][0]}, {corner_tl[1][0] + corner_br
           [1][0] - corner_bl[1][0]}};
        //use the calculated corner points to find a warped
           frame
        GetWarpedFrame finalframe = new GetWarpedFrame(frame,
           corner_tr, corner_br, corner_bl, corner_tl);
        //access the getwarped class to take the
           perspectiveTransformation matrix
        perspectiveTransformation = finalframe.getTransform();
        //call Yolodetector method. Note that Yolo is detecting
           finger coordinates on the input frame and then these
```

```
            coordinates are mapped to the warped frame
        frame = yoloDetector(frame, perspectiveTransformation);
    }

    // If missing ID is 1 (0+2+3=5)
    else if (sum == 5) {
        for (int i = 0; i < 3; i++) {
            int ID = (int) ids.get(i, 0)[0];
            Mat markerCorners = corners.get(i);
            CornerPoints pointvalues = new CornerPoints(
                markerCorners, ID, size);
            pointvalues.getMarkerCorners();
        }
        int corner_tr[][] = {{CornerPoints.markerfeatures
            [0][2][1]}, {CornerPoints.markerfeatures[0][2][2]}};
            //bottom left corner of TL marker
        int corner_bl[][] = {{CornerPoints.markerfeatures
            [2][0][1]}, {CornerPoints.markerfeatures[2][0][2]}};
            //Tl corner of BR marker
        int corner_tl[][] = {{CornerPoints.markerfeatures
            [3][1][1]}, {CornerPoints.markerfeatures[3][1][2]}};
            //TR corner of BL marker
        int corner_br[][] = {{corner_tr[0][0] + corner_bl[0][0]
            - corner_tl[0][0]}, {corner_tr[1][0] + corner_bl
            [1][0] - corner_tl[1][0]}};
        GetWarpedFrame finalframe = new GetWarpedFrame(frame,
            corner_tr, corner_br, corner_bl, corner_tl);
        perspectiveTransformation = finalframe.getTransform();
        frame = yoloDetector(frame, perspectiveTransformation);
    }

    // If missing ID is 2 (0+1+3=4)
    else if (sum == 4) {
        for (int i = 0; i < 3; i++) {
            int ID = (int) ids.get(i, 0)[0];
            Mat markerCorners = corners.get(i);
            CornerPoints pointvalues = new CornerPoints(
                markerCorners, ID, size);
            pointvalues.getMarkerCorners();
        }
        int corner_tr[][] = {{CornerPoints.markerfeatures
            [0][2][1]}, {CornerPoints.markerfeatures[0][2][2]}};
            //bottom left corner of TL marker
        int corner_br[][] = {{CornerPoints.markerfeatures
            [1][3][1]}, {CornerPoints.markerfeatures[1][3][2]}};
            //BR corner of TR marker
```

```
                int corner_tl[][] = {{CornerPoints.markerfeatures
                    [3][1][1]}, {CornerPoints.markerfeatures[3][1][2]}};
                     //TR corner of BL marker
                int corner_bl[][] = {{corner_br[0][0] + corner_tl[0][0]
                    - corner_tr[0][0]}, {corner_br[1][0] + corner_tl
                    [1][0] - corner_tr[1][0]}};
                GetWarpedFrame finalframe = new GetWarpedFrame(frame,
                    corner_tr, corner_br, corner_bl, corner_tl);
                perspectiveTransformation = finalframe.getTransform();
                frame = yoloDetector(frame, perspectiveTransformation);
            }

            // If missing ID is 3 (0+1+2=3)
            else if (sum == 3) {
                for (int i = 0; i < 3; i++) {
                    int ID = (int) ids.get(i, 0)[0];
                    Mat markerCorners = corners.get(i);
                    CornerPoints pointvalues = new CornerPoints(
                        markerCorners, ID, size);
                    pointvalues.getMarkerCorners();
                }
                int corner_tr[][] = {{CornerPoints.markerfeatures
                    [0][2][1]}, {CornerPoints.markerfeatures[0][2][2]}};
                     //bottom left corner of TL marker
                int corner_br[][] = {{CornerPoints.markerfeatures
                    [1][3][1]}, {CornerPoints.markerfeatures[1][3][2]}};
                     //BR corner of TR marker
                int corner_bl[][] = {{CornerPoints.markerfeatures
                    [2][0][1]}, {CornerPoints.markerfeatures[2][0][2]}};
                     //Tl corner of BR marker
                int corner_tl[][] = {{corner_tr[0][0] + corner_bl[0][0]
                    - corner_br[0][0]}, {corner_tr[1][0] + corner_bl
                    [1][0] - corner_br[1][0]}};
                GetWarpedFrame finalframe = new GetWarpedFrame(frame,
                    corner_tr, corner_br, corner_bl, corner_tl);
                perspectiveTransformation = finalframe.getTransform();
                frame = yoloDetector(frame, perspectiveTransformation);
            }
        }

        //if all of the markers detected
        if (ids.size(0) > 3) {
            vibro1 = 0;
            vibro2 = 0;
            fpsMeter.measure();
            int size = 4;
```

```
                //iterate over each Aruco marker
                for (int i = 0; i < 4; i++) {
                    int ID = (int) ids.get(i, 0)[0];
                    if (ID < 4) { //sometimes wrong ID numbers are detected.
                         So restricted them to <4
                        Mat markerCorners = corners.get(i);
                        //call the CornerPoints class
                        CornerPoints pointvalues = new CornerPoints(
                           markerCorners, ID, size);
                        //call the method within that class
                        pointvalues.getMarkerCorners();
                    }
                }
                //get 2D arrays of 4 corner points. Only one corner of each
                   marker is detected to get a new frame enclosed by inner
                   corners
                int corner_tr[][] = {{CornerPoints.markerfeatures[0][2][1]},
                     {CornerPoints.markerfeatures[0][2][2]}}; //bottom left
                   corner of TL marker
                int corner_br[][] = {{CornerPoints.markerfeatures[1][3][1]},
                     {CornerPoints.markerfeatures[1][3][2]}}; //BR corner of
                     TR marker
                int corner_bl[][] = {{CornerPoints.markerfeatures[2][0][1]},
                     {CornerPoints.markerfeatures[2][0][2]}}; //Tl corner of
                     BR marker
                int corner_tl[][] = {{CornerPoints.markerfeatures[3][1][1]},
                     {CornerPoints.markerfeatures[3][1][2]}}; //TR corner of
                     BL marker
                GetWarpedFrame finalframe = new GetWarpedFrame(frame,
                   corner_tr, corner_br, corner_bl, corner_tl);
                perspectiveTransformation = finalframe.getTransform();
                frame = yoloDetector(frame, perspectiveTransformation);
            }


            perspectiveTransformation.release();

            System.gc();
        }
        return frame;
    }

    @Override
    public void onCameraViewStarted(int width, int height) {
        //initialize appropriate cfg and weights files for detection
        tinyYoloCfg = getPath("yolov3-tiny_custom.cfg", this);
```

```
        tinyYoloWeights = getPath("yolov3-tiny_custom_last.weights", this);
        tinyYolo = Dnn.readNetFromDarknet(tinyYoloCfg, tinyYoloWeights);


    }
    @Override
    public void onCameraViewStopped() {
    }
    @Override
    protected void onResume() {
        super.onResume();
        if (!OpenCVLoader.initDebug()){
            Toast.makeText(getApplicationContext(),"There's a problem, yo!",
                 Toast.LENGTH_SHORT).show();
        }
        else
        {
            baseLoaderCallback.onManagerConnected(baseLoaderCallback.SUCCESS
                );
        }
    }
    @Override
    protected void onPause() {
        super.onPause();
        if(cameraBridgeViewBase!=null){
            cameraBridgeViewBase.disableView();
        }
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        if (cameraBridgeViewBase!=null){
            cameraBridgeViewBase.disableView();
        }
    }
}
```

GetWarpedFrame.java

```
package com.example.YoloDetectionFiveFingers;

import org.opencv.core.Mat;
import org.opencv.imgproc.Imgproc;
import org.opencv.utils.Converters;
import org.opencv.core.Point;
```

```
import java.util.ArrayList;
import java.util.List;

public class GetWarpedFrame {
    static Mat mRgba;
    int x1;
    int y1;
    int x2;
    int y2;
    int x3;
    int y3;
    int x4;
    int y4;
    public static double maxWidth;
    public static double maxHeight;
    public static int newHeight = 600;
    public static int newWidth = 900;

    GetWarpedFrame(Mat mRgba, int[][] corner_tr, int[][] corner_br, int[][]
        corner_bl, int[][] corner_tl) {
    //GetWarpedFrame(Mat mRgba, int[][] corner_tl, int[][] corner_tr, int
        [][] corner_br, int[][] corner_bl) {

        // corners order changed because landscape orientation is used.
            Aruco marker 3 is top left now
        this.x1 = corner_tl[0][0];
        this.y1 = corner_tl[1][0];
        this.x2 = corner_tr[0][0];
        this.y2 = corner_tr[1][0];
        this.x3 = corner_br[0][0];
        this.y3 = corner_br[1][0];
        this.x4 = corner_bl[0][0];
        this.y4 = corner_bl[1][0];
        this.mRgba = mRgba;
    }

    public Mat getTransform() {
        //find maximum width of the new warped frame
        double widthA = Math.sqrt((Math.pow((x3 - x4), 2)) + (Math.pow((y3 -
            y4), 2)));
        double widthB = Math.sqrt((Math.pow((x2 - x1), 2)) + (Math.pow((y2 -
            y1), 2)));
        maxWidth = Math.max(widthA, widthB);

        //find maximum height
```

```
        double heightA = Math.sqrt((Math.pow((x2 - x3), 2)) + (Math.pow((y2
            - y3), 2)));
        double heightB = Math.sqrt((Math.pow((x1 - x4), 2)) + (Math.pow((y1
            - y4), 2)));
        maxHeight = Math.max(heightA, heightB);

        //create a matrix of the destination warped frame
        List<Point> dstPoints = new ArrayList<>();
        dstPoints.add(new Point(0, 0));
        dstPoints.add(new Point(maxWidth - 1, 0));
        dstPoints.add(new Point(maxWidth - 1, maxHeight - 1));
        dstPoints.add(new Point(0, maxHeight-1));
        Mat dstMat = Converters.vector_Point2f_to_Mat(dstPoints);

        //create a matrix of the source warped frame
        List<Point> srcPoints = new ArrayList<>();
        srcPoints.add(new Point(x1, y1));
        srcPoints.add(new Point(x2, y2));
        srcPoints.add(new Point(x3, y3));
        srcPoints.add(new Point(x4, y4));

        Mat srcMat = Converters.vector_Point2f_to_Mat(srcPoints);

        //get the transformation matrix
        Mat perspectiveTransformation = Imgproc.getPerspectiveTransform(
            srcMat, dstMat);

        srcMat.release();
        dstMat.release();
        System.gc();

        //return the transformation matrix
        return perspectiveTransformation;
    }
}
```

fetchData.java

```
package com.example.YoloDetectionFiveFingers;


import android.os.AsyncTask;
import android.util.Log;

import java.io.BufferedReader;
```

```
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.*;
import static java.lang.System.out;

public class fetchData extends AsyncTask<String, String, String> {

    public static List<String> list = new ArrayList<String>();
    public static HashMap<String, String> dict = new HashMap<String, String
        >();
    String fetched_data;




    protected String doInBackground(String... params) {


        HttpURLConnection connection = null;
        BufferedReader reader = null;

        try {
            URL url = new URL(params[0]);
            connection = (HttpURLConnection) url.openConnection();
            connection.connect();


            InputStream stream = connection.getInputStream();

            reader = new BufferedReader(new InputStreamReader(stream));

            String line = "";

            while ((line = reader.readLine()) != null) {

                fetched_data = line;
                Log.d("Response: ", "> " + fetched_data);   //here u ll get
```

```
                whole response..... :-)

            }
            fetched_data = fetched_data.substring(1, fetched_data.length()
               -1);//remove curly brackets
            fetched_data = fetched_data.replaceAll("\"", "");
            String[] keyValuePairs = fetched_data.split(",");


            for (String s : keyValuePairs) {
                String key = s.split(":")[0];
                String value = s.split(":")[1];
                dict.put(key,value);

            }

            list = Arrays.asList(fetched_data.split(","));

            return null;


        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (connection != null) {
                connection.disconnect();
            }
            try {
                if (reader != null) {
                    reader.close();
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
        return null;
    }
}
```

CornerPoints.java

```
package com.example.YoloDetectionFiveFingers;
```

```
import org.opencv.core.Mat;
import org.opencv.core.Point;

//initialize class
public class CornerPoints {
    private static final String TAG = "OCVSample::Activity";
    public static int ID;
    static int size;
    Mat markerCorners;
    public Point[] cornerValues = new Point[4]; //this are the corner points
        of a single marker
    public static int[][][] markerfeatures = new int[4][4][3]; //here all of
        the 16 or 12 corners of 4 or 3 markers are saved

    //class constructor
    CornerPoints(Mat markerCorners, int ID, int size) {
        this.ID = ID;
        this.markerCorners = markerCorners;
        this.size = size;
    }

    //initialize method
    public void getMarkerCorners() {
        //one loop accesses 4 points of  marker. Note that each marker data
           is sent separately from the Main activity. This class deals with
            single marker.
        for (int i = 0; i < 4; i++) {
            //use the Mat of markercorners to access the corner values only.
            this.cornerValues[i] = new Point(markerCorners.get(0, i)[0],
                markerCorners.get(0, i)[1]);
            //this is not necessary and can be removed
            markerfeatures[ID][i][0] = 0;
            //use markercorners to access x and y coordinate of each of the
                4 corners of the single marker. 8 points on total
            markerfeatures[ID][i][1] = (int) this.cornerValues[i].x;
            markerfeatures[ID][i][2] = (int) this.cornerValues[i].y;

        }

        markerCorners.release();
        System.gc();

    }


}
```

OrientationManager.java

```
package com.example.YoloDetectionFiveFingers;

import android.view.OrientationEventListener;
import android.content.Context;


public class OrientationManager extends OrientationEventListener {

    public enum ScreenOrientation {
        REVERSED_LANDSCAPE, LANDSCAPE, PORTRAIT, REVERSED_PORTRAIT
    }

    public ScreenOrientation screenOrientation;
    private OrientationListener listener;

    public OrientationManager(Context context, int rate, OrientationListener
         listener) {
        super(context, rate);
        setListener(listener);
    }

    public OrientationManager(Context context, int rate) {
        super(context, rate);
    }

    public OrientationManager(Context context) {
        super(context);
    }

    @Override
    public void onOrientationChanged(int orientation) {
        if (orientation == -1){
            return;
        }
        ScreenOrientation newOrientation;
        if (orientation >= 60 && orientation <= 140){
            newOrientation = ScreenOrientation.REVERSED_LANDSCAPE;
        } else if (orientation >= 140 && orientation <= 220) {
            newOrientation = ScreenOrientation.REVERSED_PORTRAIT;
        } else if (orientation >= 220 && orientation <= 300) {
            newOrientation = ScreenOrientation.LANDSCAPE;
        } else {
            newOrientation = ScreenOrientation.PORTRAIT;
        }
```

```
        if(newOrientation != screenOrientation){
            screenOrientation = newOrientation;
            if(listener != null){
                listener.onOrientationChange(screenOrientation);
            }
        }
    }

    public void setListener(OrientationListener listener){
        this.listener = listener;
    }

    public ScreenOrientation getScreenOrientation(){
        return screenOrientation;
    }

    public interface OrientationListener {

        public void onOrientationChange(ScreenOrientation screenOrientation)
            ;
    }
}
```

KalmanFilter.java

```
package com.example.YoloDetectionFiveFingers;

import org.opencv.core.KalmanFilter; // Import Kalman filter


public class MainActivity extends AppCompatActivity implements
    CameraBridgeViewBase.CvCameraViewListener2 {

    // Kalman filter variables
    private KalmanFilter kalmanFilter;
    private Mat measurement;
    private Point lastPredictedPoint; // Store the last predicted point


    // Initialize Kalman filter in onCameraViewStarted
    @Override
    public void onCameraViewStarted(int width, int height) {

        // Kalman filter initialization
```

```
        kalmanFilter = new KalmanFilter(4, 2, 0); // 4 state variables, 2
            measurement variables
        measurement = new Mat(2, 1, CvType.CV_32F);
        // Set initial state (assuming starting at 0,0)
        kalmanFilter.statePost.put(0, 0, 0);
        kalmanFilter.statePost.put(1, 0, 0);
        kalmanFilter.statePost.put(2, 0, 0);
        kalmanFilter.statePost.put(3, 0, 0);
        // Set initial covariance matrix (adjust as needed)
        setIdentity(kalmanFilter.measurementMatrix);
        setIdentity(kalmanFilter.processNoiseCov, new Scalar(1e-4));
        setIdentity(kalmanFilter.measurementNoiseCov, new Scalar(1e-1));
        setIdentity(kalmanFilter.errorCovPost, new Scalar(1));
        lastPredictedPoint = new Point(0, 0); // Initial prediction
    }


    // Modify the yoloDetector method to apply Kalman filter
    public Mat yoloDetector(Mat frame, Mat perspectiveTransformation){


                // **Apply Kalman Filter:**
                measurement.put(0, 0, new_centerX);
                measurement.put(1, 0, new_centerY);

                Mat prediction = kalmanFilter.predict(); // Predict
                    fingertip position
                Point predictedPt = new Point(prediction.get(0, 0)[0],
                    prediction.get(1, 0)[0]);

                kalmanFilter.correct(measurement); // Update Kalman filter
                    with measurement

                // Use predictedPt (from Kalman filter) instead of
                    new_centerX and new_centerY


                // Update lastPredictedPoint
                lastPredictedPoint = predictedPt;

    }

}
```