Computer Vision for visually impaired children

ABSTRACT

The gaming industry has seen outstanding advancements, yet there remains a significant gap in accessible gaming options for visually impaired individuals, particularly children. This project addresses this gap by developing a computer vision-based application designed to make the card game UNO accessible to blind children. Utilising OpenCV for image processing, TensorFlow/Keras for machine learning, and Pyttsx3 for text-to-speech (TTS) capabilities, the application identifies UNO cards in real-time, providing immediate auditory feedback to the player. The program's modular design supports both webcam-based live recognition and image uploads, while a hand gesture recognition feature enhances interactivity. The key challenges, such as overfitting in model training, difficulties in colour detection under varied lighting conditions, and system crashes post voice output, were systematically addressed through data augmentation, transfer learning, and iterative debugging. The project highlights the potential of integrating computer vision and machine learning to create inclusive gaming experiences and underscores the importance of ongoing innovation in this field to further enhance accessibility and interactivity for visually impaired users.

I. INTRODUCTION AND BACKGROUND

The gaming industry has undergone a transformative evolution, transitioning from traditional tabletop games to sophisticated digital platforms, significantly advancing accessibility and playability [1]. However, despite these advancements, there is a huge gap in the market, the development of games specifically for visually impaired individuals, particularly children who are blind. In England and Wales alone, approximately 37,000 children are blind [2]. Addressing this issue presents not only a technological challenge but also an opportunity for profound social enrichment. As a solution, this paper proposes a novel application that leverages computer vision and machine learning, utilising OpenCV and Python, to revolutionise the gaming experience for visually impaired children. The goal is not merely to replicate traditional board games in a digital format but to enhance these games in ways that promote inclusivity and interactivity. As visual accessibility remains a significant challenge, particularly in games that heavily rely on visual cues, such as card games. Therefore, this project will focus on improving these types of games starting with the classic game UNO, which exemplifies the challenges of recognising colours and numbers, while making strategic decisions based on what the player sees [3]. For blind individuals, engaging in such games can be difficult without modifications, which may include Braille cards or assistance from a sighted person [4]. However, these traditional methods often fall short in providing an equitable and enjoyable gaming experience. By critically analysing the potential and challenges encountered during this project's progression, this report underscores the synergy between engineering innovation and social responsibility [5][6].

In addition to explaining the development process, this report synthesizes relevant scientific literature, evaluating the current landscape of assistive gaming technologies and positioning this project within that context. It also discusses the emerging market demand for accessible gaming solutions and projects future developments in the field [7]. This paper aims to provide a comprehensive view in accessible gaming, highlighting both the technological and societal impacts of these ground-breaking developments.

1.1. Introduction to Accessibility Challenges in Games

The importance of making games like UNO accessible is needed as playing tabletop games is a useful tool for social interaction and cognitive development, particularly for children. Games like UNO help develop skills in strategic thinking, pattern recognition, and social interaction, all of which are critical to a child's overall development [8]. For blind children, the inability to participate fully in such activities can lead to social exclusion and missed opportunities for cognitive growth [9]. Given the significance of this issue, there has been a growing interest in leveraging modern technology to bridge the accessibility gap in gaming. Specifically, advancements in computer vision (CV) and machine learning (ML) offer promising avenues for making games like UNO accessible to blind children. By using these technologies, it is possible to create systems that can recognise and interpret visual game elements, and provide real-time feedback to visually impaired players, enhancing their ability to participate in the game and improving the overall gaming experience.

1.2. The Role of Technology in Enhancing Accessibility

Technology has long been a tool for enhancing accessibility in various domains, from physical mobility to communication. In the context of gaming, technology can play a crucial role in overcoming the barriers faced by individuals with disabilities. For visually impaired individuals, technology can be used to translate visual information into auditory or tactile formats [10], making it possible to interact with games that they were previously unable to play. In recent years, computer vision and machine learning have emerged as key technologies in this effort. Computer vision, which involves the use of algorithms to interpret and understand visual information, can be used to identify and describe the visual elements of a game. Machine learning, particularly deep learning, can further enhance this process by enabling systems to learn from data and improve their accuracy over time [11].

In the context of UNO, these technologies can be used to develop systems that recognise the cards being played and provide appropriate audio feedback to the player. For example, a camera could capture an image of a card, and a computer vision algorithm could identify the card’s colour and number. This information could then be relayed to the player through an auditory cue, such as a voice reading out the card’s details. Such a system would not only allow blind children to play UNO independently, but it could also be integrated into more complex gaming environments, where multiple players with varying levels of visual ability can participate together. This would help create a more inclusive gaming environment, where all players can engage with the game on an equal footing.

However, implementing such a system has its challenges. The accuracy of computer vision algorithms can be affected by various factors, such as lighting conditions, the quality of the camera, and the design of the cards themselves. Moreover, providing real-time feedback requires the system to process information quickly and efficiently, which can be computationally demanding. Despite these challenges, the potential benefits of using technology to enhance accessibility in gaming are significant. Not only does it provide blind children with the opportunity to engage in social and cognitive activities that are important for their development as demonstrated by the University of Texas [12]. To further extend, it also promotes inclusivity by allowing them to participate in shared activities with their sighted peers.

II. OPENCV IN ENHANCING CARD GAME ACCESSIBILITY

2.1. OpenCV Basics and Capabilities

OpenCV is one of the most widely used computer vision libraries, offering a vast array of functionalities that can be applied to various image and video processing tasks. Its open-source nature makes it accessible to developers and researchers, allowing for extensive customisation and integration with other tools and technologies. The library includes multiple modules, each dedicated to specific tasks in image processing and analysis. For instance, the core module provides essential functions for matrix operations, image processing, and basic arithmetic operations. The imgproc module offers tools for image filtering, edge detection, contour finding, and more, which are critical for tasks like object recognition and segmentation.

In the context of enhancing card game accessibility, several key functionalities of OpenCV can be particularly useful:

1. Image Pre-processing: Before any recognition task, images need to be pre-processed to enhance the quality and make it suitable for further analysis. OpenCV offers a variety of pre-processing techniques, including noise reduction (e.g., Gaussian blurring or Filters), contrast enhancement, and thresholding. These techniques are crucial when working with images captured in varying lighting conditions or with different levels of noise.

2. Edge Detection and Contour Finding: Edge detection algorithms, such as the Canny edge detector, are fundamental in identifying the boundaries of objects within an image. In card recognition, detecting the edges of the cards is the first step in isolating them from the background. Contour finding algorithms can then be used to outline these edges, making it easier to identify the shape and orientation of the cards.

3. Colour Space Conversion: UNO cards are distinguished by their vibrant colours, making colour recognition a crucial aspect of the game. OpenCV supports various colour spaces, such as RGB, HSV, and Lab. By converting images to the HSV colour space, where colour information is separated from intensity, it becomes easier to accurately detect and classify the colours of the cards under different lighting conditions.

4. Template Matching: Recognising the specific symbols and numbers on UNO cards can be achieved through template matching, where a predefined template of a card (or a part of it) is compared against regions in the captured image. OpenCV provides functions for template matching, allowing for the identification of the exact location of known patterns within an image.

5. Feature Detection and Matching: Beyond template matching, OpenCV also offers more advanced feature detection methods, such as SIFT (Scale-Invariant Feature Transform) and ORB (Oriented FAST and Rotated BRIEF). These techniques allow for the detection of key points and the matching of these points between different images. This is particularly useful in recognising cards from different perspectives or under varying conditions.

Once the features are detected, they can be matched against a database of known features from the UNO cards. This approach is more robust than template matching, particularly when dealing with images taken from different angles or under varying conditions. Feature matching can help in recognising cards even when they are partially obscured or when only a portion of the card is visible.

Challenges in Image Processing:
While the techniques described above are powerful, they are not without challenges. For instance, varying lighting conditions can significantly impact the accuracy of colour detection and template matching. Shadows, reflections, and changes in light intensity can cause the system to misinterpret the colours or shapes of the cards. Additionally, the quality of the camera and the resolution of the images play a crucial role in the effectiveness of these techniques [13].

Moreover, real-world conditions often present challenges such as occlusions, where a part of the card is covered by another object, or where the card is partially out of the frame. In such scenarios, feature detection and matching techniques become essential, as they can still identify the card based on the visible features. To address these challenges, it is important to implement robust pre-processing steps, such as adaptive thresholding, which adjusts the threshold dynamically based on the local characteristics of the image, and histogram equalisation, which enhances the contrast of the image. These pre-processing techniques can help improve the accuracy of the subsequent image processing tasks.

2.2. Real-Time Card Recognition Using OpenCV

The ultimate goal of using OpenCV in enhancing the playability of UNO for blind children is to enable real-time card recognition. This requires not only accurate image processing techniques but also efficient algorithms that can process and interpret images quickly enough to provide immediate feedback to the player.

III. MACHINE LEARNING FOR ENHANCED ABILITY AND PLAYABILITY.

3.1 Overview of Machine Learning Models

Machine learning has revolutionised various domains by enabling systems to learn from data and improve their performance over time [14]. In the context of enhancing the adaptability and playability of UNO for blind children, machine learning offers powerful tools for recognising and interpreting the visual elements of the game. While traditional image processing techniques, such as those provided by OpenCV, are essential for basic tasks like edge detection and colour identification, machine learning, particularly deep learning, allows for more sophisticated pattern recognition and decision-making processes. Traditional machine learning algorithms, such as Support Vector Machines (SVMs), decision trees, and k-nearest neighbours (k-NN), have been widely used in image recognition tasks [15]. These algorithms are effective for classification problems where the goal is to assign a label to an input based on its features. For example, an SVM could be trained to classify images of UNO cards into different categories (e.g., red cards, green cards, number cards, action cards). However, traditional machine learning algorithms typically require manual feature extraction, where the relevant features (such as edges, textures, or shapes) are identified and extracted from the images before classification. This process can be time-consuming and may not capture the full complexity of the visual data, especially when dealing with high-dimensional inputs like images.

Deep learning addresses these limitations by automatically learning the features from the data. This is particularly advantageous in image recognition tasks, where the features may be complex and difficult to define manually. Convolutional Neural Networks (CNNs) is a type of deep learning model that has become the de facto standard for image-related tasks due to their ability to capture spatial hierarchies in images.

A CNN consists of multiple layers of neurons, each responsible for detecting different features of the image. The initial layers typically detect simple features like edges and textures, while deeper layers identify more complex patterns, such as shapes and objects [16]. This hierarchical approach allows CNNs to learn robust representations of the input data, making them highly effective for tasks like card recognition in UNO. In the context of making UNO accessible to blind children, a CNN could be trained on a large dataset of UNO card images. The network would learn to recognise different cards based on their visual characteristics, such as colour or number. Once trained, the CNN could be used in real-time to classify cards as they are played, providing immediate feedback to the player.

Training a deep learning model from scratch requires a large amount of labelled data and significant computational resources. However, transfer learning offers a way to leverage pre-trained models that have been trained on large datasets, such as ImageNet, to perform specific tasks with less data and computation [17]. In transfer learning, the pre-trained model's knowledge is transferred to a new task by fine-tuning the model on a smaller, task-specific dataset [18]. For instance, a CNN pre-trained on ImageNet could be fine-tuned on a smaller dataset of UNO card images. This approach significantly reduces the amount of data and time required to train the model while still achieving high accuracy. By using transfer learning, developers can create effective card recognition systems without needing to collect and label thousands of images from scratch. This makes the development process more accessible and feasible, particularly in educational and research settings where resources may be limited.

3.2. Adaptive Learning Systems for Personalized Gameplay

While accurate card recognition is crucial for making UNO accessible to blind children, it's equally important to ensure that the system adapts to the individual needs and preferences of the player. Adaptive learning systems, powered by machine learning, can personalise the gaming experience by adjusting the feedback [19].

Children of different ages have varying cognitive abilities, preferences, and learning styles. An adaptive learning system can take these differences into account by customising the gameplay experience for each age group.
For younger children, the system might focus on providing simpler instructions and more frequent feedback to help them understand the game. The gameplay could also be slower paced, allowing them more time to make decisions and respond to feedback. On the other hand, for older children or those with more experience, the system could introduce a faster-paced gameplay to keep them engaged. The feedback could also be more nuanced, encouraging them to think strategically and develop their skills further. By adapting to the needs of different age groups, the system ensures that UNO remains an enjoyable and educational experience for all players, regardless of their age or cognitive abilities.

IV. CASE STUDIES AND PRACTICAL IMPLEMENTATIONS

Several research projects and practical implementations have explored the use of machine learning for enhancing accessibility in gaming, particularly for visually impaired individuals. These case studies offer valuable insights into the challenges, successes, and potential future directions of using machine learning to improve the playability and adaptability of card games like UNO.

4.1. Case Study 1: Real-Time Detection and Recognition System for Set Card Game

Overview and Objectives

This case study focuses on developing a real-time detection and recognition system for the card game “Set”. The objective of this system was to enable quick identification and classification of cards from the “Set” game during live video feeds. The system aimed to perform accurate card detection, feature extraction, and classification efficiently enough to allow real-time gameplay. By leveraging computer vision techniques and machine learning classifiers, the goal was to create a robust tool that could handle the complexities of the game while maintaining high accuracy and speed.

Technical Implementation

The system's implementation involved several stages: card segmentation, feature extraction, and classification. The process began with capturing video frames containing the cards using a standard camera. To isolate the cards from the background, OpenCV was employed to detect edges and contours in the image. This step was crucial for identifying the card boundaries and performing perspective transformation to standardise the card orientation. Once the cards were segmented, the system extracted features relevant to the game's four card attributes: colour, shape, number of shapes and shading. The feature extraction was done using a method called scanlines, which involved sampling horizontal and vertical pixel lines from the segmented card images. For each feature, a specific classifier was trained. The classifiers included support vector machines (SVMs) with different kernel functions tailored to the specific feature being analysed. For example, the colour classifier utilised pixels from the second horizontal scanline, chosen based on their saturation and brightness in HSV colour space, while the shape classifier used the distances between hit positions of scanlines to determine the shape on the card. The system also included a fully connected neural network (FCNN) to distinguish whether a segmented image truly represented a card from the Set game.

User Testing and Results

The system was tested using a dataset of 935 images, capturing various lighting conditions and card orientations. The classifiers demonstrated high accuracy, ranging from 97% to 100% on the test dataset, with the number of shapes classifier performing particularly well. Despite this success, real-world application tests revealed some limitations. For instance, the colour classifier struggled under low-light conditions, often misclassifying purple cards due to their darker appearance. Additionally, the fill classifier had difficulty distinguishing between outline and striped fills when cards were farther from the camera, a result of reduced image resolution. Nevertheless, the system's overall performance was impressive, with the classifiers proving fast and reliable enough for real-time use. The FCNN classifier effectively filtered out non-card objects, although it showed a bias towards non-card classifications in certain scenarios [21].

Lessons Learned

This case study highlighted the effectiveness of combining traditional computer vision techniques with machine learning classifiers for real-time card recognition. The simplicity and speed of the feature extraction and classification methods were key to achieving the desired real-time performance. However, the limitations encountered, particularly in challenging lighting conditions and with certain card features, underscore the need for further improvements. Future enhancements could involve expanding the training dataset to improve classifier robustness and exploring deep learning approaches that might offer greater accuracy and flexibility. The study also emphasised the importance of selecting the right features and classifiers to balance performance with computational efficiency in real-time applications.

4.2. Case Study 2: Real-Time Playing Card Recognition System Using Contour Shape Matching

Overview and Objectives

This case study focuses on the development of a real-time recognition system for playing cards, designed to assist visually impaired individuals in playing card games. The system uses mobile phone images to identify playing cards based on contour shape matching techniques. The primary objective was to develop a system capable of recognising the suit and number of each card in real-time, regardless of the card’s orientation, thus enhancing accessibility for visually impaired users and enabling fairer gameplay through digital card storage.

Technical Implementation

The system’s implementation follows a sequence of image processing steps, starting with capturing a 24-bit true-colour image of the card using a mobile phone. The image is then converted into a grayscale image, reducing it to a 256-color scale. Subsequently, the image undergoes binarization, transforming it into a binary image consisting only of black and white pixels. This simplification is essential for isolating the card's contours from the background. Next, the system applies contour shape matching using pre-stored templates of the card suits (hearts, spades, diamonds, and clubs) and ranks (Ace to King). The core of the recognition process relies on Hu invariant moments, which are mathematical descriptors that remain unchanged under image transformations such as rotation and scaling. This characteristic ensures that the system can accurately recognise cards even if they are rotated or positioned differently within the image. The process involves extracting the region of interest (ROI) from the grayscale image, focusing on the card's corners where the suit and number are typically located. After identifying the contours in the ROI, the system matches these contours with the pre-loaded templates using the Hu invariant moments. The template that produces the smallest distance in the moment space is considered the best match, and the corresponding suit and number are then identified.

User Testing and Results

The system was tested with a series of images captured under different conditions, including variations in lighting and shooting angles. The images were taken against a dark background to improve contrast and ensure clearer contour detection. The results showed that the system successfully identified the correct suit and number of cards in most cases, demonstrating the robustness of the Hu invariant moment-based contour matching approach. However, the testing also revealed some limitations. Recognition accuracy could be affected by variations in the shooting environment and the angle at which the card was photographed. For example, certain numbers, such as 2, 3, and 5, occasionally caused confusion during recognition, as did similar-looking cards like 6 and 9. Despite these challenges, the system performed reliably overall, delivering results with minimal processing time [22].

Lessons Learned

This case study illustrates the effectiveness of using Hu invariant moments for contour shape matching in real-time playing card recognition. The approach proved to be both efficient and relatively simple to implement, making it a suitable choice for mobile applications aimed at assisting visually impaired users. The key strength of the system lies in its ability to maintain recognition accuracy despite changes in the card’s orientation, thanks to the rotational invariance of the Hu moments. However, the study also emphasised the sensitivity of the recognition process to environmental factors such as lighting and camera angles. Future work could focus on enhancing the system's robustness by incorporating additional image pre-processing steps or exploring more advanced recognition algorithms. Moreover, expanding the template database and refining the contour matching process could help address the occasional misidentifications observed during testing. Therefore, this case study demonstrates the potential of combining classic image processing techniques with contour shape matching to create accessible and fair card gaming experiences, while also identifying areas for further refinement and improvement.

4.3. Case Study 3: Hand Gesture Recognition Interface for Visually Impaired and Blind People

Overview and Objectives

This case study explores the development of a hand gesture recognition system aimed at improving human-computer interaction for visually impaired and blind individuals. The system uses a touchpad interface to recognise hand gestures, simplifying the input process and reducing the need for additional hardware components. The primary objective was to create an accessible and efficient input solution that can be easily learned and used in everyday scenarios, allowing visually impaired users to interact with wearable computing systems more effectively.

Technical Implementation

The system was built around the use of a Cirque Corporation TSM9925 USB touchpad, which acts as a standard USB mouse. Data was captured by intercepting USB reports on a Linux system, recording timestamped events of movements along the X and Y axes. This raw data was then processed and labelled for machine learning purposes. The touchpad data was converted into a stream of events, with each event corresponding to a gesture made by the user. To train and test the recognition system, a dataset of gestures based on the graffiti handwriting system was collected. Participants were asked to enter each graffiti symbol multiple times, generating a dataset of 3250 instances. This dataset was then split into training and test sets, with 66% of the data used for training. The remaining data was used to evaluate the performance of various machine learning algorithms implemented using the WEKA data mining software. Several machine learning algorithms were tested, including Naive Bayes, Hidden Markov Models (HMM), Neural Networks, C4.5 Decision Trees, Support Vector Machines (SVM), and K-Nearest Neighbour (KNN). Each algorithm was evaluated based on its ability to accurately recognise the graffiti symbols from the touchpad data. The input data was pre-processed in multiple ways, including normalisation, interpolation, and filling missing values, to determine the optimal configuration for each algorithm.

User Testing and Results

The system's performance was evaluated through a qualitative user study involving five participants, three of whom had prior experience with gesture recognition devices. The study revealed that all tested algorithms achieved high recognition rates, with Neural Networks showing the highest overall accuracy. However, the training time for Neural Networks was significantly longer compared to other algorithms. Participants without prior experience in gesture recognition still achieved recognition rates above 94%, demonstrating the system's ease of use and the effectiveness of the chosen machine learning models. However, certain letters, such as B, D, and G, were more challenging to recognise accurately across all algorithms due to their similar shapes and drawing sequences, leading to occasional misclassifications [23].

Lessons Learned

This case study highlights the potential of hand gesture recognition as a practical input solution for visually impaired and blind users interacting with wearable computing systems. The results demonstrate that even with minimal prior experience, users can quickly achieve high recognition accuracy with the system, making it a viable alternative to more traditional input methods like keyboards or voice recognition. One of the key takeaways is the importance of selecting appropriate machine learning algorithms and pre-processing techniques to balance accuracy and processing time. While Neural Networks provided the best results, their longer training times may not be suitable for all applications. Simpler algorithms like Naive Bayes and SVM also performed well, offering a good trade-off between speed and accuracy. However, the study also identified specific challenges, such as the misclassification of certain similar-looking symbols. Future iterations of the system could benefit from addressing these issues, possibly by refining the gesture set or incorporating additional features to distinguish between challenging symbols more effectively.

4.4. Comparative Analysis of Case Studies

Technical Approaches and Challenges

The three case studies illustrate diverse approaches to solving accessibility and interaction challenges using computer vision and machine learning techniques. In the first case study, the “Set” card game recognition system employed a three-stage process involving card segmentation, feature extraction, and classification using support vector machines (SVMs). This approach leveraged traditional image processing techniques combined with machine learning to deliver high accuracy in real-time. However, challenges arose in low-light conditions, where colour classification became less reliable, and in distinguishing similar-looking shapes when the cards were distant from the camera. On the other hand, the second case study, which focused on playing cards recognition via contour shape matching, used Hu invariant moments to ensure that recognition was unaffected by the card's orientation. This method was effective in maintaining recognition accuracy across various rotations and translations of the cards. Nonetheless, the system faced difficulties in environments with inconsistent lighting or when dealing with cards that had similar contour shapes, leading to occasional misclassifications. Finally, the third case study did not involve image recognition but rather the interpretation of touchpad gestures. This system applied several machine learning algorithms, including neural networks and SVMs, to process gestures based on the graffiti alphabet. The primary challenge in this case was ensuring high recognition accuracy across different users with varying levels of familiarity with gesture-based interfaces. The study highlighted that while neural networks provided the best recognition rates, they required significant computational resources and longer training times.

User Experience and Feedback

All three case studies prioritised user accessibility, with the goal of enabling independent interaction for visually impaired users. The “Set” card game system and the playing card recognition system both aimed to make card games more accessible by providing real-time feedback. In the “Set” game system, this feedback allowed users to make strategic decisions during gameplay, though the need for improved performance in suboptimal conditions was evident. The playing card recognition system similarly focused on making the gaming experience accessible, particularly by aiding visually impaired users in identifying cards through tactile feedback.

The hand gesture recognition system, designed for wearable computing, offered a different kind of user interaction. It enabled visually impaired users to input text or commands through simple gestures, minimising the need for external hardware. User feedback indicated that the system was intuitive and could be learned quickly, even by those without prior experience with gesture recognition. However, the study also pointed out the need for refining the gesture set to avoid misrecognition of similar gestures.

Lessons for Future Development

The comparative analysis of these case studies yields several key lessons for the development of accessible interaction systems.

Adaptability and Robustness: Future systems should be designed to handle a variety of environmental conditions, such as changes in lighting or camera angles. Algorithms that adapt in real-time to these variations can significantly enhance the user experience and ensure consistent performance.

Simplification of Interaction: Reducing the complexity of user interaction, as seen in the hand gesture recognition system, can lead to more intuitive and user-friendly interfaces. Simplified gestures or commands that require minimal learning can make systems more accessible to users with diverse abilities.

Accuracy vs. Efficiency: Balancing accuracy with computational efficiency is crucial, especially in real-time applications. While neural networks may offer superior accuracy, their resource-intensive nature could be a limitation in environments where quick response times are essential.

User-Centred Design: Involving users in the design process and gathering feedback on their experiences can lead to more effective and satisfying solutions. Customisation options that allow users to tailor the system to their needs, whether through feedback modes or input methods are essential for broader usability.

4.5. Application of Case Study Insights to UNO Card Recognition Project

The three case studies provide valuable insights that can be directly applied to the development of the UNO card recognition system, particularly in areas such as image processing, contour detection, model training, gesture recognition, and user-centred design.

Firstly, the application of image processing and contour detection in case study 1 could be used for the creation of the UNO card recognition. In these studies, OpenCV was used effectively to pre-process images, including tasks such as resizing, normalisation, and edge detection, which were essential for isolating and identifying key features of the cards. Furthermore, the use of contour detection, particularly in the playing card recognition case study, where Hu invariant moments were employed, highlights the importance of accurately identifying card boundaries and shapes. For the UNO project, utilising similar image processing and contour detection methods will be crucial in ensuring that the system can reliably detect and identify UNO cards, especially when dealing with varying card orientations and environmental conditions.

The training of neural network models for recognising specific attributes of the cards, such as colour and numbers, is similar to the “Set” card recognition study. While the Set study employed support vector machines (SVMs), the underlying principle of training specialised models to handle distinct aspects of the recognition task is consistent. This approach ensures that the system can achieve high accuracy in recognising the unique features of each card. For the UNO project, training separate neural networks for colour and number recognition, as seen in these case studies, would enhance the system’s ability to accurately identify and classify the cards under different scenarios.

Incorporating gesture recognition as part of the user interaction is relevant. The study demonstrated the effectiveness of using simple gestures to trigger specific actions, reducing the need for complex user inputs. Applying a similar approach in the UNO project, where hand gestures are used to activate voice feedback, can simplify the interaction process and make the system more intuitive for users. This is particularly important in creating an accessible interface for visually impaired users, as it allows for seamless and efficient communication between the user and the system.

Finally, the emphasis on user-centred design and real-time feedback found across all three case studies is critical to the success of the UNO card recognition system. Ensuring that the system provides immediate auditory feedback, as well as incorporating customisable options for users, is essential in creating an inclusive and user-friendly experience. The studies highlight the importance of tailoring feedback mechanisms to meet the needs of diverse users, particularly those with visual impairments. Implementing similar principles in the UNO project will help ensure that the system not only meets technical requirements but also delivers a meaningful and accessible user experience.

V. METHODOLOGY

5.1. Project Initialisation

Objective

The primary objective of this project was to develop a computer vision program capable of recognising UNO cards through either a webcam feed or uploaded images. The program was also designed to provide real-time voice feedback when a card was recognised. This approach aims to enhance the accessibility of board games for visually impaired children by providing auditory feedback during gameplay.

Tool Selection

To achieve this objective, the following tools and libraries were selected:

- Programming Language: Python, chosen for its robust support in machine learning and wide range of libraries.
- Libraries: OpenCV for image processing tasks such as resizing, normalisation, and edge detection. TensorFlow/Keras for developing and training the neural network models. MediaPipe for real-time hand gesture recognition. Pyttsx3 for implementing text-to-speech (TTS) functionality.

5.2. Model Development

Data Collection

A dataset comprising various UNO card images was collected and labelled according to card categories (colour and action). The dataset included a mix of standard cards and special action cards to ensure the model could generalise across different card types.

Model Training

Two distinct neural network models were developed and trained using TensorFlow/Keras:

- Colour Model: To recognise the colour of the card.
- Action Model: To identify the specific action or number on the card.

Both models were trained on the labelled dataset and subsequently saved for integration into the main program. The training accuracy and loss were monitored over several epochs, as shown in the training dataset graph, which helped in identifying overfitting and other performance challenges.

5.3. Program Design and Structure

Process Flow

The development of the card recognition software followed a methodical process to ensure accuracy and reliability. The process flow, as illustrated in ***Figure 1***, summarises the card recognition process. The program starts by loading the trained model and setting up the necessary parameters for card detection. Once the game begins, the program captures images of the cards, using computer vision to isolate and identify the cards against varied backgrounds.

Modular Design

The program was designed with a modular structure to enhance clarity and reusability. Key functions implemented included:

- main_menu(): The primary interface for user interaction.
- upload_menu(): Allows users to upload images for card recognition.
- live_menu(): Enables real-time recognition using a webcam feed.
- process_frame(): Handles the image processing and model prediction for each frame.
- draw_card_contour(): Draws contours around detected cards in images.
- speak_result(): Converts the recognized card information into speech.

User Interaction

A text-based user interface was developed, offering users options to either upload an image or use the live camera feed for card recognition. The program handled inputs efficiently to ensure smooth transitions between different modes (upload, live, exit).

5.4. Image Processing and Contour Detection

Image Processing

OpenCV was utilised for image pre-processing tasks, which included resizing, normalisation, and edge detection. By using refined HSV colour ranges, the system processed the captured image to detect the card's colour and number, adapting to varying lighting conditions.

Contour Detection

A custom function was implemented to detect and draw contours around the detected card in an image. This function was further enhanced by adding a feature that draws a specific trigger area on the card, indicating where the user’s hand gesture should be for the program to trigger voice feedback.

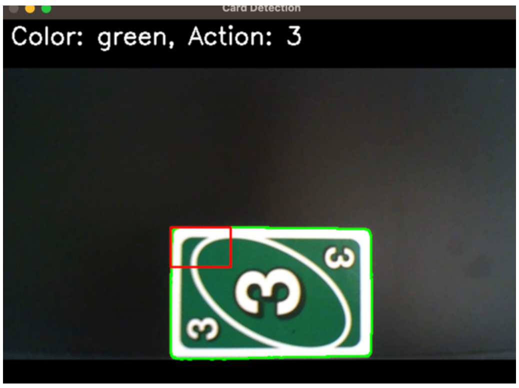


Image 1 – Shows the card contour detection and the trigger area for gesture recognition.

5.5. Hand Gesture Recognition

Hand Detection Integration

MediaPipe was integrated into the program to detect hand landmarks in real-time. This feature was crucial for implementing gesture-based interactions.

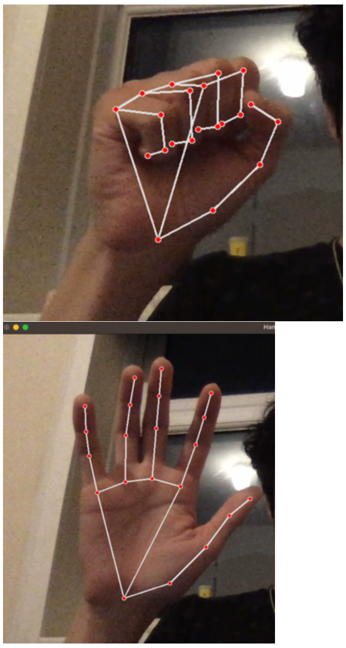


Image 2 & 3 – Shows the hand being detected.

Gesture Recognition

The program was designed to recognise when the user's thumb and index finger were close together near the card's corner. If this gesture was detected within the trigger area, the program would activate the TTS function to speak the card's details.

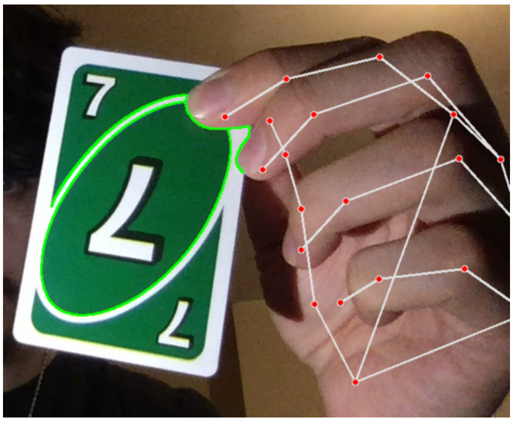


Image 4 – Shows the hand detection and the card detection.

5.6. Voice Feedback Implementation

Text-to-Speech (TTS) Integration

Pyttsx3 was used to convert recognised card details into speech, providing auditory feedback to the user. This integration was essential for ensuring that the program could verbally communicate the card details to visually impaired users.

```
Processing frame in live mode.
Predicting card in live mode.
Running prediction on card.
1/1 [==============================] - 0s 59ms/step
1/1 [==============================] - 0s 61ms/step
Predicted card color: green, action: 7
Drawing contours on card.
Hand gesture recognized: index and thumb close.
Hand gesture detected.
Speaking result: The card is green with action 7
```

Image 5 – Shows the process of providing auditory feedback.

Threading for TTS

To ensure the voice feedback did not block the main program's execution, threading was implemented. This allowed the TTS engine to run concurrently with the main program, enabling continuous operation without interruption.

5.7. Program Testing and Debugging

Testing Each Mode

Both the upload and live modes were rigorously tested to ensure that card recognition was accurate, and the voice feedback functioned as intended. Unit and system tests were conducted to ensure software reliability. Metrics such as accuracy, response time, and user satisfaction were measured to evaluate the program's effectiveness.

Colour Classification Report:

	precision	recall	f1-score	support
blue	1.00	0.67	0.80	3
red	1.00	1.00	1.00	3
green	1.00	1.00	1.00	3
yellow	0.75	1.00	0.86	3
accuracy			0.92	12
Macro avg	0.94	0.92	0.91	12
Weighted avg	0.94	0.92	0.91	12

Debugging Issues

Several issues were addressed during the debugging phase:

- Unexpected Exits: Issues related to program flow, such as unexpected exits after voice feedback, were resolved.

- Speech Blocking: The program was modified to ensure that the main loop continued running after delivering voice feedback, preventing any blocking issues.

5.8. Further tests

Visual Inspection and HSV Tuning

This image series demonstrates a visual inspection to assess what the program is focusing on during the card detection process. The primary objective of this inspection was to fine-tune the HSV (Hue, Saturation, Value) colour range settings for optimal card detection.

Image 1: Shows the initial mask applied to the card. The mask isolates the card from the background, but some noise is present, which could lead to inaccurate colour detection. The main task here is to remove as much noise as possible to ensure the model can focus solely on the card's colour.

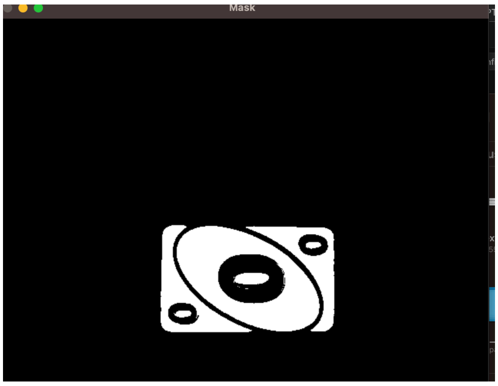


Image 2: Depicts the manual adjustment process for refining the HSV values. The sliders in the interface are used to tweak the hue, saturation, and value thresholds until the card's colour is correctly isolated. This step is crucial as it allows for the identification of the best HSV values for detecting the colour blue under different lighting conditions.

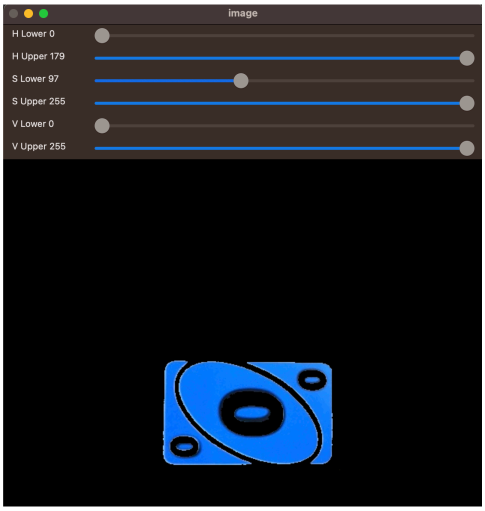
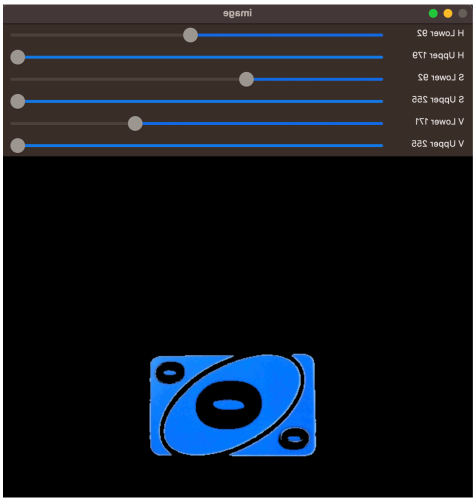
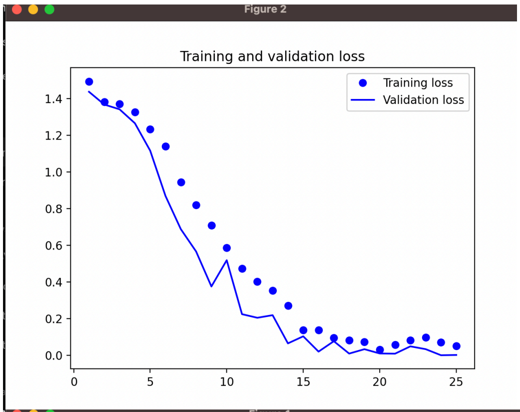


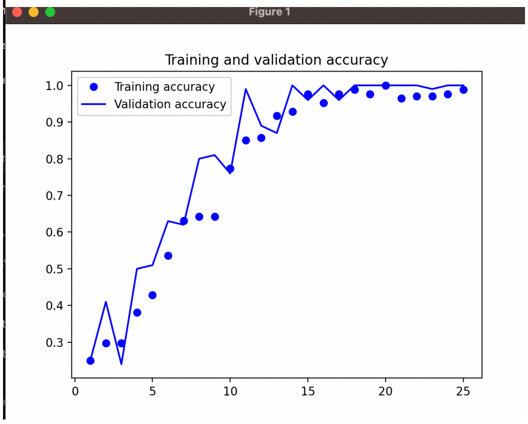
Image 3: Illustrates the result after adjustments. The blue colour of the card is now clearly isolated, with minimal noise. This refined mask will serve as the input for the CNN (Convolutional Neural Network) used in colour classification.



CNN for Colour Classification

After refining the HSV values, a CNN was employed to classify the colours of the UNO cards. The CNN model was trained on the adjusted dataset, which now had more accurate colour representations due to the refined HSV values.

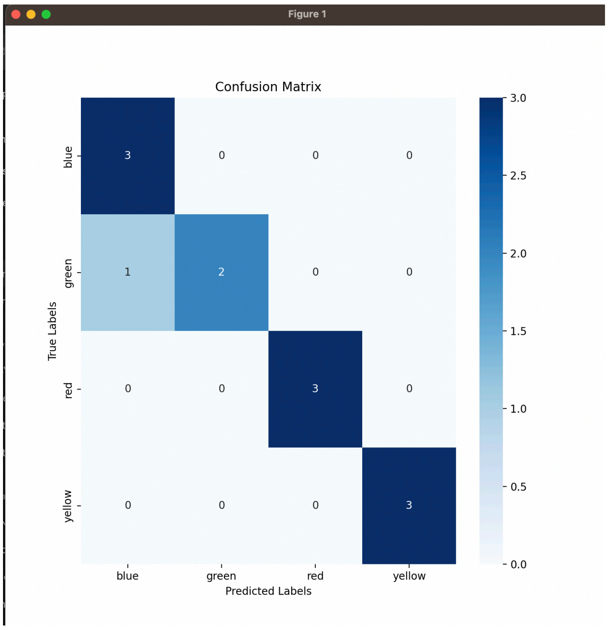




The use of a confusion matrix was integral in understanding the specific areas where the model underperformed. For instance, if red cards were often misclassified as blue, this issue would be evident in the matrix. The classification report further provided metrics for each class, such as precision, recall, and f1-score, offering insights into potential biases or challenges the model faced in recognizing certain colours.

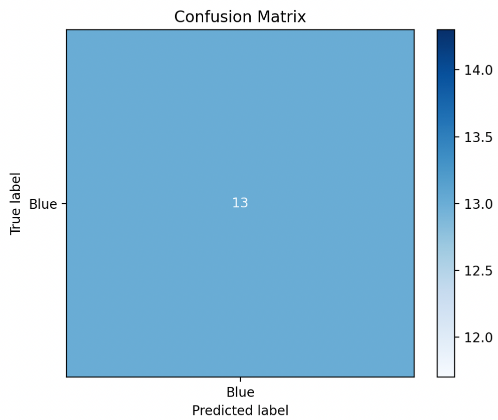
Classification Report:

	precision	recall	f1-score	support
blue	0.75	1.00	0.86	3
red	1.00	0.67	0.80	3
green	1.00	1.00	1.00	3
yellow	1.00	1.00	1.00	3
accuracy			0.92	12
Macro avg	0.94	0.92	0.91	12
Weighted avg	0.94	0.92	0.91	12



This report indicates that while the model performed well overall, there were still some challenges, particularly with distinguishing green cards, which had a slightly lower recall score. The insights gained from these metrics guided further refinements in data augmentation and model tuning to improve accuracy across all card colours.

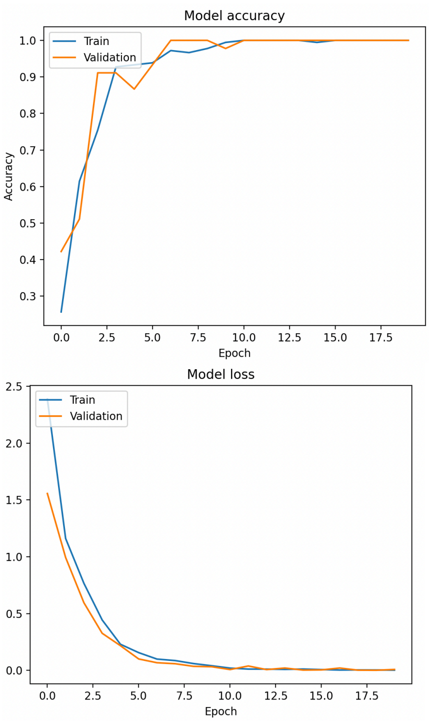
Overfitting and Training on Single Colour



To further investigate the model's performance, additional tests were conducted by training the CNN on a single colour, in this case, blue. The results were as follows:

```
1/1
[=============================
=] - 0s 379ms/step
Image: b9.jpg - Predicted Label: Blue
Image: b8.jpg - Predicted Label: Blue
Image: bT.jpg - Predicted Label: Blue
Image: b5.jpg - Predicted Label: Blue
Image: b4.jpg - Predicted Label: Blue
Image: b6.jpg - Predicted Label: Blue
Image: b7.jpg - Predicted Label: Blue
Image: bR.jpg - Predicted Label: Blue
Image: b3.jpg - Predicted Label: Blue
Image: b2.jpg - Predicted Label: Blue
Image: bS.jpg - Predicted Label: Blue
Image: b0.jpeg - Predicted Label: Blue
Image: b1.jpg - Predicted Label: Blue
```

The results indicate that while the model successfully identified all the training images as blue, it struggled to generalise to new, unseen data, which is a classic sign of overfitting. The model memorised the training data instead of learning to generalise, highlighting the need for more diverse training data or possibly more sophisticated regularisation techniques during training.



Graph shows the final model accuracy and loss.

5.8. User Interface Enhancements

Visual Feedback

To provide better visual feedback to users, a zoomed-in view of the detected card was added to the interface.

Gesture Trigger Area

A red rectangle was drawn on the card's contour to indicate where the user should make the gesture for voice feedback, enhancing the program's usability.

5.9. Final Testing and Validation

Comprehensive Testing

Final testing was conducted across various environments to validate that the program performed as expected in different scenarios. Testing also included isolating relevant features using image masks to ensure accurate recognition, as shown in the example mask applied to a card image.

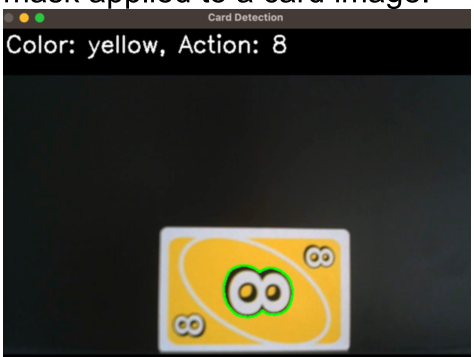


Image 6 & 7 – Numbered 8 and 7 being recognised.

Performance Optimization

The program was optimised to reduce latency in the live camera mode, ensuring smooth and real-time card recognition and feedback.

VI. CHALLENGES, LIMITATIONS AND FUTURE RESEARCH

6.1 Errors and Challenges Encountered

Program Crashing After Voice Output

Issue: Initially, the program experienced unexpected crashes following the voice output. This was particularly problematic after the Text-to-Speech (TTS) function was executed.

Solution: The issue was resolved by refining the program's flow and implementing threading to ensure that the voice output didn't block the main execution loop. This allowed the program to continue running smoothly after the TTS execution.

Voice Output Not Triggering in Live Mode

Issue: There were instances where the voice output failed to trigger during live mode, particularly when the hand gesture was not correctly recognised or when the TTS engine was not properly invoked.

Solution: This issue was resolved by improving the hand gesture detection logic. The program was adjusted to ensure that the TTS engine was correctly triggered only when the hand gesture was in the correct position near the card's corner. This enhancement ensured that the voice feedback was reliably activated.

Colour Detection Challenges

Issue: During the recognition of a yellow "7" card, the program initially misidentified other areas in the background or similar colours as part of the card. This resulted in incorrect recognition and misclassification.

Solution: The issue was addressed by refining the colour range detection parameters and implementing a contour-based filtering mechanism. This allowed the program to focus more accurately on the card's contour and colour, reducing the chances of false positives.

Image 8 – Shows the colour detection problem.

Distance-Related Recognition Issues

Issue: The program sometimes struggled to accurately recognise the card's number when the card was held too far from the camera. This led to delays in recognition and sometimes misclassification, as demonstrated in the image with the green "5" card.

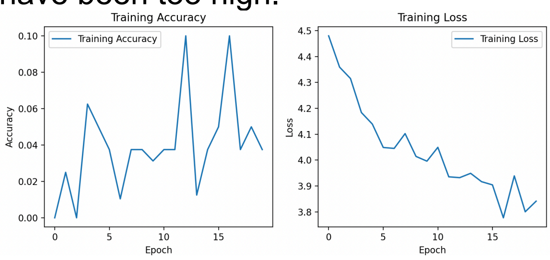
Image 9 – Shows number 5 being misread.

Solution: While the issue of distance remains a limitation, the program's accuracy was improved by optimising the image processing pipeline. This included enhancing the contour detection and resizing mechanisms to make the program more resilient to variations in distance.

Image 10 – Shows number 5 being recognised.

Model not learning

During the initial training phase, as illustrated in the graph, the model struggled to learn effectively. The training accuracy fluctuated significantly with little overall improvement, and the training loss, although decreasing, exhibited an erratic pattern. This lack of consistent progress suggested several potential issues, including a dataset that was too small or imbalanced, possible overfitting, or a learning rate that might have been too high.



The unpredictable behaviour in accuracy and the slow reduction in loss indicated that the model was not generalising well and was potentially memorising the training data without learning meaningful patterns. These challenges were likely compounded by a limited and possibly non-diverse dataset, which would have further hindered the model's ability to learn effectively.

To address these issues, several strategies were implemented. Data augmentation was employed to artificially increase the size and diversity of the dataset, which helped the model to see a broader range of examples during training. Transfer learning was also utilised by fine-tuning a pre-trained model, which provided a solid foundation of learned features that the model could build upon. Additionally, hyperparameter tuning was conducted to optimise key parameters, such as the learning rate, and regularisation techniques like dropout were applied to prevent overfitting. These solutions collectively helped the model to overcome the initial learning challenges, resulting in more stable and consistent improvements in both training accuracy and loss.

6.2. Limitations

The accuracy of gesture recognition varies depending on lighting conditions and the positioning of the user's hand. The model's performance is limited by the size and diversity of the dataset, which could lead to overfitting or underfitting in certain scenarios. Furthermore, the distance the card is being held was another limitation as the program can get confused and think it is a different number. Moreover, covering part of the number with the finger can also reduce the precision.

6.3. Future Research

- Dataset Expansion: Collecting a larger and more diverse dataset could improve the model's ability to generalise across various conditions and card types.
- Advanced Gesture Recognition: Implementing more sophisticated gesture recognition techniques or integrating additional sensors could enhance the program's interactivity and reliability.
- Real-time Performance: Further optimisations could be made to improve the program's responsiveness and reduce latency in real-time recognition.

VII. CONCLUSION

The development and implementation of a computer vision program for the game UNO, aimed at enhancing accessibility for visually impaired children, represents a significant advancement in the field of assistive technology. This project successfully integrated various cutting-edge technologies, including OpenCV for image processing, TensorFlow/Keras for machine learning, and Pyttsx3 for text-to-speech capabilities, to create a system that allows blind children to independently play UNO and have a better experience. Throughout the project, several key achievements were made. The program was able to accurately detect and recognise UNO cards in real-time, providing immediate auditory feedback to the player. This was made possible through meticulous data collection, model training, and the application of sophisticated image processing techniques. The use of CNNs for colour classification further enhanced the system's accuracy, although challenges such as overfitting and difficulties in generalising to new data were encountered and addressed through various strategies like data augmentation and transfer learning.

The program's modular design and user interface enhancements ensured that it was not only functional but also user-friendly, allowing for easy interaction even for those with limited technical knowledge. The integration of hand gesture recognition added an additional layer of interactivity, making the system more intuitive and accessible. However, the project was not without its challenges. Issues such as program crashes after voice output, difficulties with colour detection under certain conditions, and distance-related recognition problems highlighted the complexities involved in developing such a system. These challenges were addressed through iterative testing, debugging, and optimisation, leading to a more robust and reliable final product. Looking forward, there are several avenues for future research and development. Expanding the dataset and improving the diversity of training data could help the model better generalise across different scenarios. Additionally, exploring more advanced gesture recognition techniques and further optimising the system's real-time performance could significantly enhance the user experience.

In conclusion, this project not only demonstrates the potential of computer vision and machine learning in creating accessible gaming solutions but also underscores the importance of ongoing research and development in this field. By continuing to refine and expand upon this work, we can create even more inclusive and engaging gaming experiences for visually impaired individuals, ultimately contributing to greater social inclusion and cognitive development.

VIII. FIGURE

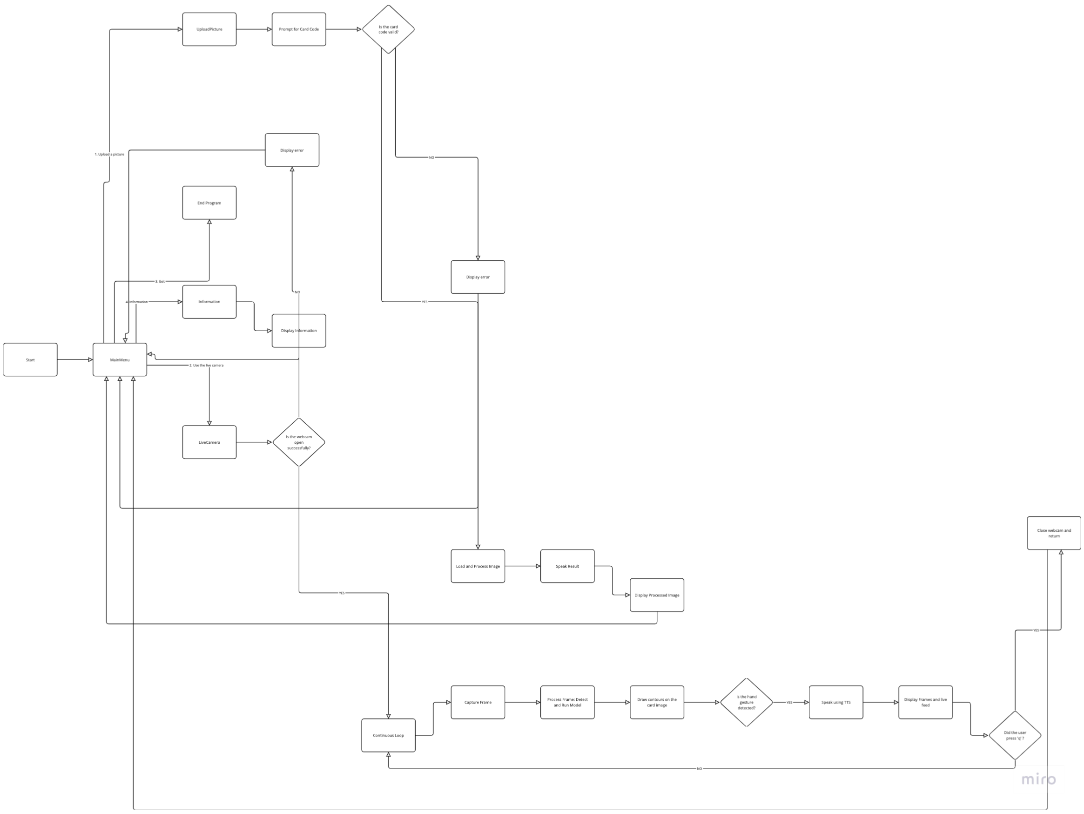


Figure 1. Flowchart

IX. REFERENCES

[1] F. da Rocha Tomé Filho, P. Mirza-Babaei, B. Kapralos, and G. Moreira Mendonça Junior, "Let’s Play Together: Adaptation Guidelines of Board Games for Players with Visual Impairment," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Glasgow, UK, May 2019, p. 631.

[2] RSBC, "Key facts," 2022. Available at: https://www.rsbc.org.uk/get-our-help/keyfacts/#:~:text=There%20are%20an%20estimated%2037%2C000,be%20diagnosed%20with%20sight%20loss

[3] E. D. Demaine, M. L. Demaine, R. Uehara, T. Uno, and Y. Uno, "UNO Is Hard, Even for a Single Player," in *Fun with Algorithms (FUN 2010)*, Lecture Notes in Computer Science, vol 6099, P. Boldi and L. Gargano, Eds. Springer, Berlin, Heidelberg, 2010, pp. 1-10. doi: 10.1007/978-3-642-13122-6_15

[4] Y. King Penny Wan, "Barriers for people with disabilities in visiting casinos," *International Journal of Contemporary Hospitality Management*, vol. 25, no. 5, pp. 660-682, 2013. doi: 10.1108/IJCHM-Jul-2012-0112

[5] P. Konieczny, "Book Review: Stewart Woods, Eurogames: The Design, Culture and Play of Modern European Board Games," *International Sociology*, vol. 32, no. 6, pp. 671–673, 2017. doi: 10.1177/0268580917731940

[6] R. J. Potter, "Encounters: Two Studies in the Sociology of Interaction.

Erving Goffman," *American Journal of Sociology*, vol. 68, pp. 125–126, 1962. doi: 10.1086/224621

[7] C. G. Bernardo, A. Mori, T. R. Cotta Orlandi, and C. G. Duque, "Multimodality by electronic games as assistive technology for visual disabilities," in *Proceedings of the 2016 1st International Conference on Technology and Innovation in Sports, Health and Wellbeing (TISHW)*, Vila Real, Portugal, Dec. 2016, pp. 1-8. doi: 10.1109/TISHW.2016.7847790

[8] J. Park, J. Shin, J. Lee, and J. Jeong, "Inter-Brain Synchrony Pattern Investigation on Triadic Board Game Play-Based Social Interaction: An fNIRS Study," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 2923-2932, 2023. doi: 10.1109/TNSRE.2023.3292844

[9] I. Casas, "Social Exclusion and the Disabled: An Accessibility Approach," *The Professional Geographer*, vol. 59, no. 4, pp. 463–477, 2007. doi: 10.1111/j.1467-9272.2007.00635.x

[10] Z. Wang and J. Ben-Arie, "Conveying visual information with spatial auditory patterns," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 6, pp. 446-455, Nov. 1996. doi: 10.1109/89.544529

[11] N. Sebe, *Machine Learning in Computer Vision*, vol. 29, Springer Science & Business Media, 2005.

[12] R. Baker, K. Ramos, and J. Turner, "Game design for visually-impaired individuals: creativity and innovation theories and sensory substitution devices influence on virtual and physical navigation skills," *Irish Journal of Technology Enhanced Learning*, vol. 4, no. 1, pp. 36-47, 2018. doi: 10.22554/ijtel.v4i1.51

[13] G. Hollinger, N. Ward, and E. C. Everbach, "Introducing computers to blackjack: Implementation of a card recognition system using computer vision techniques," Colby College, Waterville, 2003.

[14] I. El Naqa and M. J. Murphy, "What is machine learning?" in *Machine Learning in Radiation Oncology: Theory and Applications*, Springer International Publishing, 2015, pp. 3-11.

[15] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR)*, vol. 9, no. 1, pp. 381-386, 2020.

[16] P. P. Shinde and S. Shah, "A review of machine learning and deep learning applications," in *Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Pune, India, Aug. 2018, pp. 1-6. doi: 10.1109/ICCUBEA.2018.8697433

[17] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, IGI Global, 2010, pp. 242-264.

[18] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43-76, 2020. doi: 10.1109/JPROC.2020.3004555

[19] G. P. Bharathi, I. Chandra, D. P. R. Sanagana, C. K. Tummalachervu, V. S. Rao, and S. Neelima, "AI-driven

adaptive learning for enhancing business intelligence simulation games," *Entertainment Computing*, vol. 50, pp. 100699, 2024. doi: 10.1016/j.entcom.2023.100699

[20] K. Zutis and J. Hoey, "Who’s counting? real-time blackjack monitoring for card counting detection," in *Proceedings of the International Conference on Computer Vision Systems*, Berlin, Heidelberg: Springer, 2009, pp. 354-363.

[21] H. Ditrih, S. Grgić, and L. Turković, "Real-Time Detection and Recognition of Cards in the Game of Set," in *Proceedings of the 2021 International Symposium ELMAR*, Zadar, Croatia, 2021, pp. 161-164. doi: 10.1109/ELMAR52657.2021.9550996

[22] X. Hu, T. Yu, K. Wan, and J. Yuan, "Poker card recognition with computer vision methods," in *Proceedings of the 2021 IEEE International Conference on Electronic Technology, Communication and Information (ICETCI)*, Changchun, China, 2021, pp. 11-15. doi: 10.1109/ICETCI53161.2021.9563607

[23] M. Modzelewski and E. B. Kaiser, "Hand Gesture Recognition Interface for Visually Impaired and Blind People," in *Proceedings of the 2012 IEEE/ACIS 11th International Conference on Computer and Information Science*, Shanghai, China, 2012, pp. 201-206. doi: 10.1109/ICIS.2012.56