INTERACTIVE LEARNING SYSTEM FOR KIDS

2022-254

Final Report

Dhanushka Dilshan Bandara Ekanayaka

IT19138732

Supervisor : Ms. Namalie Walgampaya

Co Supervisor : Ms. Hansi De Silva

B.Sc. (Hons) Degree in Software Engineering

Department of Information Technology

Sri Lanka Institute of Information Technology

January 2022

Declaration of the Candidate & Supervisor

We declare that this is our own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
Ekanayaka M.G.D.D.B	IT19138732	

The supervisor/s should certify the proposal report with the following declaration.

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Signature of the supervisor: Date

Abstract

Advanced technology-based applications are widely used for early childhood education in this era of information technology. It is a well-known fact that children belonging to new generations tend to explore new virtual learning experiences in a fun-seeking environment. Along with that avenue, this study outlines the creation and implementation of a novel mobile application called "interactive learning system for kids" with the goal of improving key skills related to primary education. As one of the main components of the entire research project, this paper presents the development of self-learning applications for children to improve their alphabetical knowledge and skills. The component of drawing the alphabet has two main features: identifying and predicting mistakes in hand-drawn letters. Then demonstrate the correct way to write the English alphabet. The app allows the learner to freely write letters on a given screen area in order to test and evaluate the hand drawing using deep learning, convolutional neural network (CNN), and transfer learning (VGG16) with the utilization of two datasets. Detecting the handwritten alphabet is driven by three different models using CNN based on the transfer learning (VGG16) method. The accuracy rate for the final outcome of the model is more than 80%. A mobile learning application teaches the English alphabet while incorporating fun learning and arousing interest among children. This research effort would provide great learning experience and teaching aids to primary-level students without the need for manual supervision.

Acknowledgement

I would like to express our deepest and sincere gratitude to our research supervisor Ms.Namalie Walgampaya (Senior Lecturer -Faculty of computing) for the invaluable guidance, direction, and encouragement given throughout this research project. It was a great privilege and honor to work and study under her guidance. We would like to especially thanks for her friendly assistance and great sense of humor. We would also like to thank Ms. Hansi De Silva (Lecturer - Faculty of computing) who guided us throughout the research as the Co-supervisor.

Also, we would like to mention our module leader Mr. Jagath Wikramarathne (Senior Lecturer -Faculty of computing) for the special advice and guidance which were given to complete our research successfully. We appreciate the guidance and motivation that was given to us throughout the study. The authors are grateful to Mr. Malika Liyanage for his assistance in gathering the real-world dataset for our research.

Finally, we would like to thank our parents, colleagues, and all the staff and non-staff members of the SLIIT faculty of computing, Sri Lanka Institute of Information Technology who gave their blessings to complete this research project successfully.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AHWAS…………………………………...Automated Handwriting Analysis System

AR………………………………….……………………...Augmented Reality

MNIST………….....Modified National Institute of Standards and Technology dataset

OCR……………………………………………………Optical Character Recognition

CNN…………………………………….………..Convolutional Neural Network

Con2D…………………………………………………..2D Convolutional Layer

1. INTRODUCTION

1.1 Background and Literature Review

Writing abilities are significant for everyone's learning process and also professional lives as one of the main modes of communication. The traditional method of writing alphabets in all languages mostly relies on a pencil, pen, chalk on paper, or a blackboard. However, with rapid growth and changes in today's world, technology has become an influential factor in the field of education, particularly in primary education.

When considering primary education, it is the bedrock of development for every child. As a result, primary education is the first step in formal education that provides children with the most important and fundamental skills. Basically, primary or else elementary education focuses on enhancing the literacy and numeracy skills of students Therefore, it is typically designed to provide fundamental skills in reading, writing, and mathematics.

In the traditional education system, the primary education process is totally driven by manual supervision. Specially, COVID-19 pandemic created space between children and manual supervision by teachers, which had a significant impact on the education process of primary students. Parents might not have sufficient time or knowledge to assist to enhance primary level educational concepts. and also Along with the current technological advancement, commonly referred to as the "jet age," the rate at which children are beginning to use sophisticated devices, primarily to play mobile games, has increased.

Therefore, today's generation is more interested in learning from an interactive computer or mobile application–based activities than conventional educational methods. As a result of that, students, parents, and educational administrators are also looking for technological solutions to improve engagement with unique education methods..Computer or mobile device aided education tools help youngsters understand the learning process by using audio-visual assistance.

But a limited number of education systems have developed as automatic self learning platforms. By addressing the mentioned issues, this interactive learning system provides an innovative real-time learning experience for children that does not require any manual supervision. As a result, the purpose of this study was to develop a software application that would help children aged 2 to 8 improve their alphabet writing skills.

Basically, his application enables users to freely write letters on any given area of the screen in order to correctly recognize drawings as English letters. The next step is to identify the flaws in the fundamental letter drawing, and the last step is to provide an example of the proper approach to drawing the letter based on the application. Furthermore, this system would act as a self-learning platform to children to improve alphabetical knowledge and skills.

This chapter of critical literature review provides an overview and evaluation of available scholarly sources in the subject area of information technological application and systems that have been conducted related to alphabet handwriting.

Experimental studies and inventions of handwriting and letter recognition began two decades back. At the beginning, **L. D. Harmon et al** introduced a conventional approach based on statistical pattern recognition to identify the shape of the letter in 1972. When closer examination of the literature over a long period of time reveals that studies on alphabet learning have expanded into more technical aspects.

Izzat Fauzi Bin Hamidi et al aimed to develop a mobile application as an exciting and interactive way of learning the basic Jawi alphabet for kids using augmented reality technology [2]. This application enables AR technology, which is different than a regular application. Using this method can makes AR beneficial for education. And this combining with the narrator and sound will provide a better understanding of the kids that want to learn Jawi. Moreover, the study conducted by **Yusuf Perwej et al,** figured out the use of neural networks for developing a system that can recognize hand-written English alphabets [3]. Here each English alphabet is represented by the binary format that can help as a simple feature extraction system using neural network and image pre-processing. This research has tested all English alphabets with several Handwriting styles. And successfully recognized the alphabets with an average accuracy of 82.5%.

Furthermore, another experimental research study done by **Ammar H. Safar et al,**, scrutinized the effectiveness of using augmented reality (AR) applications (apps) as a teaching and learning tool when instructing kindergarten children in the English alphabet in the State of Kuwait [4]. and also, this study compared two groups: experimental, taught using AR apps, and control, taught using traditional face-to-face methods. In addition, this study concludes with relevant proposals and recommendations regarding the implementation of AR technology in education and suggests undertaking further studies on this interesting topic.

Besides, **Vikram Kamath et al,** have been carried out the behavioral prediction of a person through automated handwriting analysis. In this study handwriting is analyzed through Image Processing in MATLAB [5]. The behavioral pattern of the person is predicted from the above traits of handwriting. The developed system identifies handwriting closely with real-time processing and involves less image preprocessing. And the system is in good agreement with more than 80%. In addition, a new method is proposed for automated behavioral analysis using Automated Handwriting Analysis System (AHWAS).

According to that recent study done by **Aizan, N.L. et al** conducted his research on handwriting arabic words using tablet [6]. So, authors have evaluated tablet based handwriting and paper based hand writing using Samsung Galaxy Note smartphones. In 2016, **Géraldine Jordan et all** created the application named “LetterSchool" with the intention of covering the educational needs of primary level students [7]. That application assists handwriting with the aid of both visual and guided ways. One of the weaknesses of the " LetterSchool" was that there was no way to assess students' writing abilities.

Another author from turkey created web software that used Optical Character Recognition (OCR) as an evaluator for handwriting. In this system, children can trace letters on a tablet or a touch-screen device and then be tasseled by the system itself. Furthermore, this application demonstrated the fundamental steps to teaching turkey letters one by one. [8]

B. R. Maxim et al was focused on learning through gaming concepts by elevating the next level of educational tools influenced by information technology [9]. As a result, the authors created an immersive gaming application by combining emerging technology, computer animation, and simulation.

Another researcher **Loey, Mohamed et al** created a web application name as "Arab Kid Tutor" with a strong focus on the Arabic alphabet hand writings [10]. The authors used fuzzy logic to evaluate the hand drawn letter based on stroke count and stroke order. This research also only focuses on only for the identifying the letter and providing visual aids for the elementary level students. Another author from the same country created a guiding app for writing Arabic letters. In this app, learners are allowed to write over the give alphabetic character. Furthermore, this system extended their teaching process by completing simple words that were segmented and tested by a complex machine learning model [11].

As a summary of literature in the alphabet learning subject area, different authors all over the world have used various technological methods with the goal of developing an interactive alphabet learning system. According to the author, **Steven Sybenga et al** [12] used image processing, the MNIST dataset, and the Naive Bayes machine learning algorithm to design and implement the application.

The great author **Yanikoglu, B. et al** [13] also created a computer-based system for teaching and assessing the ability to write Turkish basic letters. It also includes modules for handwriting and arithmetic, which help elementary school students improve their academic skills. Handwriting recognition technologies rely heavily on machine learning.

A similar system has been created by **Iwayama, N. et al** [14] with Kanji letters. At that time, handwriting inventions had progressed to the checking stage. So, this system included an online handwriting recognition feature to check the handwriting.

According to the progress of this research, it revealed that automatic checking saved time and enhanced the enthusiasm and motivation to learn and try out to write more letters.

In comparison to the mobile applications and web-based systems developed, **Hu, Z. et al** [15] created a much more advanced intelligent tutor tool focused on Chinese. The system can detect and determine whether or not a handwritten letter is correct. Not only has it detected the sequence, the relationship, and the stroke production. Furthermore, authors used relational graph matching to detect handwriting errors, as well as a pruning strategy to reduce computational time.

Furthermore, one of the effective mobile applications called "Handwriting Without Tears" developed by **Kristi Hape OTD et al** [16] by paying high attention to grade 1 students' handwriting. However, this application did not focus on testing student handwritten letters. The study provided preliminary evidence that using the curriculum from their app in first-grade classrooms is effective in this dynamic educational environment.

Moving on to the most recent mobile and desktop applications available in the Google Play store and Apple App Stores that demonstrate handwriting in various alphabets in different corners of the world. The applications such as Kids Learn Bangla Alphabet, Hatekhori, Writing Wizard, iTrace, are more focused on tracing and guiding children to write letters in a professional manner. When carefully reviewing all of the selected apps, the fact that is identified is that they do not have any evaluating feature to monitor the letter writing done by learners.

Overall evaluation of the number of scholarly evidence and also existing mobile apps available has clearly highlighted that only a few studies could expand their effort to guide lesson writing with on-screen testing features. There is a vast gap in identifying mistakes in handwritten letters in those applications. Based on that, according to the best of our knowledge, no advanced applications or research studies have developed features to properly evaluate hand-drawn letters in real time.

1.2 Research Gap

A closer look at the literature reveals that several mobile applications for learning the Alphabet for kids are available worldwide, with slightly similar functionalities and features. Nonetheless, most previously developed applications allow children to draw a letter on a provided letter-shaped image or recognize handwriting. The majority of prior applications have provided alphabet letter images first and then instructed children to draw the letter on the provided image.

However, according to the present author's perspective, those types of methods did not allow children to attain proper knowledge about letters. Hence, drawing on the previously provided image would not allow children to learn for themselves. Overcoming the above limita1tions, the current study aims to explore new methods for teaching and enhancing kids' writing skills on the alphabet by themselves.

Based on that, the present author intends to cover an unstudied area by developing an alphabet hand drawing learning application that provides significantly different learning outcomes than previously developed applications. In this study, innovative methods will be used by concentrating on each stage of kids' alphabet learning methods to give the best accurate result.

Furthermore, this application expands its innovative hand-drawn procedure, which includes predicting whatever kids are drawing on the screen and identifying mistakes in the basic drawing, in order to show the correct way to draw the letter by the software itself. Not only that, but the application is showing kids the way of writing a good professional letter with the intention of providing kids with a better understanding.

According to that, this study makes novel contributions by investigating a new alphabet learning system based on electronic hand-drawn procedures, which will provide an opportunity for kids to learn about the alphabet in a magnificent and memorable way.

2. RESEARCH PROBLEM

Nowadays globalization is affecting more on humans, forcing stressful and competitive life. Allocating more time by parents to gain economic advantages, on the other hand, a big salary is essential in order to achieve the life goals. As an example, if we take a family, parents are more responsible for earning money than attaining other social and personnel needs. Sometimes their children can be neglected by them unfortunately due to this developing competition. In the present society, parents have limited time to allocate to their children. Therefore, the limited time which parents allocated for their children should be sustainable. But most of the time that does not happen. A way to spend this limited time with children sustainably is by providing good education since childhood. For instance, according to my point of view, priority should be given to providing education to their children as education is the key to modern society is more requested. So, giving more priority to give better education is more important through their limited time frame.

Most experts throughout the world are identified “Education” as the massive wealth of a person to achieve life goals successfully. Since childhood, the basement of education should be accorded by parents in order to attain the advanced result. primary education is an extensive topic related to education. primary education should provide the learner with opportunities to acquire literacy, numeracy, creativity, and communication skills basically. Without having a better primary education, children can be get lost of having next step, secondary education.

Presently, due to digitalization parents have more sources to acquire knowledge of providing good primary education to their kids. Furthermore, kids are more quickly following these sources and they are interested in learning through technological equipment such as smartphones, laptops. During a short period of time, parents can teach more to their kids. As above mentioned, a key factor of primary education is providing literacy. There are several apps to learn about Alphabet which is the basic literacy level of nurseries. But most are focused on only watching. Often kids are interesting to learn while drawing. Drawing becomes interesting and easy to understand when introduced with alphabets. Drawing is very important for children. Drawing improves wrist movement and hand-eye coordination, which will make writing letters and numbers easier later. Most of the time children have the ability to grab most skills including literacy between the ages of 3-5 years.

Drawing through learning is more effective than other learning methods for kids. So in this research, I will provide an Alphabet learning method to learn letters by him or herself with less engaging of parents. So through this innovative method parents can utilize their limited time effectively and kids can learn about the alphabet very interesting way.

3. RESEARCH OBJECTIVES

This research was conducted on the topic of Interactive Learning System for Kids which was designed to increase the learning efficiency of Kids. The end users of this application are primary age students. Existing systems and applications in this field have a number of drawbacks. As a result, our research provides a method for overcoming these obstacles and developing a system that allows users to use the services on offer in a comfortable manner. Therefore, current research is divided into four main components.

1. Identify hand-drawn letters, predict the letter if it is incomplete, and suggest corrections if the drawn letter is wrong.
2. Identify and classify hand drawn shapes and arts that drawn using shapes and give grades for those sketches.
3. Identify and classify hand drawing objects with colors
4. Generate image captions using image context

3.1 Main objectives

The primary goal of this research is to develop an interactive learning system to improve the knowledge and skills of primary-level students. Along with that, the authors expected to create an interactive learning platform that provides teaching aids to primary-level students without the need for manual supervision. Therefore, The researchers' goal is to effectively integrate advanced information technologies and early key childhood education activities such as letters, images, and shape-based activities.

3.2 Specific objectives

In order to reach the main objective, the four specific objectives stated in the present research.

01) To develop an application that can correctly identify and predict hand-drawn English letter in order to demonstrate a professional writing style from each letter in the alphabet.

According to this component, giving focus to several approaches to identify the hand-drawn alphabet and this sub objective has a combination of different features. Identifying the hand-drawn alphabet letters is very helpful to give the knowledge of letters. The DL powered platform with Transfer Learning needs to identify the letters after the drawing is done. Normally we are looking to get the best result by predicting the letter if the letter is incomplete and checking the mistakes.

The accurate detection of hand-drawn letters is very important for improving the kid’s skills and knowledge. However, nowadays automatic learning methods are needed to improve the knowledge of the kids. So, we need to identify the hand-drawn letters very accurately to get the best result. This component will take real-time processing to identify the letters.

02) Identify and classify hand drawn shapes and arts that drawn using shapes and give grades for those sketches.

Using this specific objective, the shape knowledge of primary ages students has been tried to improve. In this component a surface has provided for students to draw shapes. After finishing the drawing student will be able to see the drawn

sketch is correct or not. If the drawn sketch in correct, student will be able to see a grade for the drawing. If the drawn shape is incorrect the correct shape is displayed for students. Not only this, from this component art skills of students also are being developed. Students can draw flowers, houses and stars using basic shapes. After completes the drawing students will be able to see correctness of the drawing. If the drawing is not complete, the system will provide a "Need to be Improved" message. Then students can easily correct those mistakes and learn fast.

03) Identify and classify hand drawing objects with colors

Using this specific objective, the drawing and painting skills of primary ages students have been tried to improve. In this component a surface has been provided for students to draw some objects (Apple, Banana, carrot) and color them. After finishing the drawing student will be able to see if the drawn sketch is correct or not. If the drawn shape is incorrect the correct shape is displayed for students. In this component color knowledge of students also is being improved. As an example, when drawing an apple after drawing an apple, the student should color that drawn apple from red color. Therefore, drawing skills and color knowledge both are improving from this application.

04) Generate image captions using image context

In this component when an image provides to application a caption will generate according to the features and objects that existing on the image. The student must write a simple sentence watching that image. After that, Written sentence by the student is matching with the generated sentence by the system. If both are matching, the student has written the sentence correctly. If both did not match, the written sentence by the students is incorrect. From this component students can improve their sentence writing knowledge rapidly.

4. METHODOLOGY

This chapter explains the special steps of actions and methodologies that were used to develop the mobile application and its features.The methodology chapter for this study consists of the following key components: system architecture, application design, model creation, software solution, requirement gathering and analysis, and finally commercialization.

4.1 System Architecture

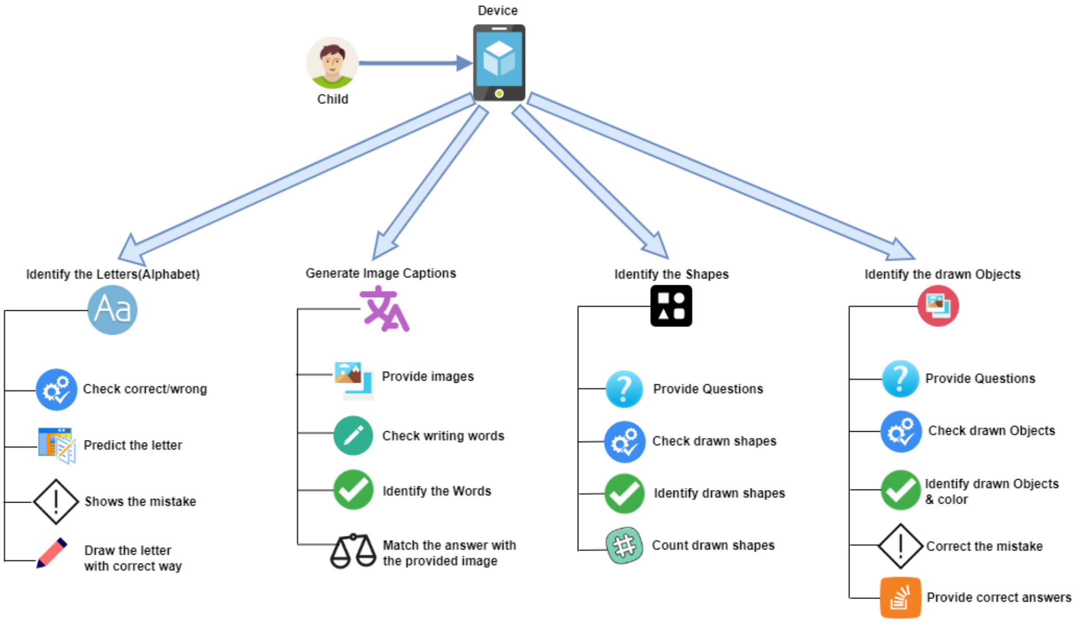

Figure 4.1.1: Overall System Diagram

This interactive learning system for kids incorporated four distinct categories of primary activities to create and design this application. As a consequence of this, the key features of this program are designed to provide assistance with the following activities: writing letters, drawing shapes, drawing images, and recognizing images. As per Figure 4.1.1 it has illustrated the main components of this entire study with key functions of each component briefly.

The hand drawn alphabet letters component first provides the option to draw the alphabet letters. then provide a space for you to draw a letter When students draw an alphabet letter, the system should recognize it as a prediction. The drawn letter was then compared to the correct letter by the system. If a student draws a letter incorrectly or incompletely, the system predicts the alphabet letter that the student was attempting to draw. If there is an error in the drawn letter, the system will detect it. The system's suggestions for correcting those errors and make a correct letter for the system's errors.

the identify shapes component, Kids have access to a surface on which they can draw various shapes. When the drawing is finished, the Kids can determine whether or not the sketch that was drawn is accurate. then displayed the grade for the drawing they have submitted if the sketch they have drawn is accurate. When a Kids has incorrectly drawn a shape, displaied of the correct shape. In addition to this, The Kids can draw basic shapes to draw things like flowers, houses, and stars. After finishing the drawing, examine it and determine whether or not it is accurate. The "Need to be Improved" message will be displayed by the system if the drawing is in any way incomplete.

A surface has been provided in the hand drawn objects using colors component for students to draw and color some objects (Apple, Banana, Carrot). After completing the drawing, the kids determine whether or not the drawn sketch is correct. If the drawn shape is incorrect, the kids can see the correct shape. Kids’ color knowledge is also improved in this component.

When an image is provided to the application through generate image caption using image context component, a caption will be generated automatically according to the characteristics and objects that are present on the image. The student is required to compose a brief sentence while viewing that picture. After that, the Kids’ written sentence is compared to the system's generated sentence to see if there are any differences. If they are consistent with one another, the Kids has correctly written the sentence. If they did not agree with one another, then the Kids' written sentence contains an error. Kids can experience dramatic growth in their ability to write effective sentences by focusing on this component.

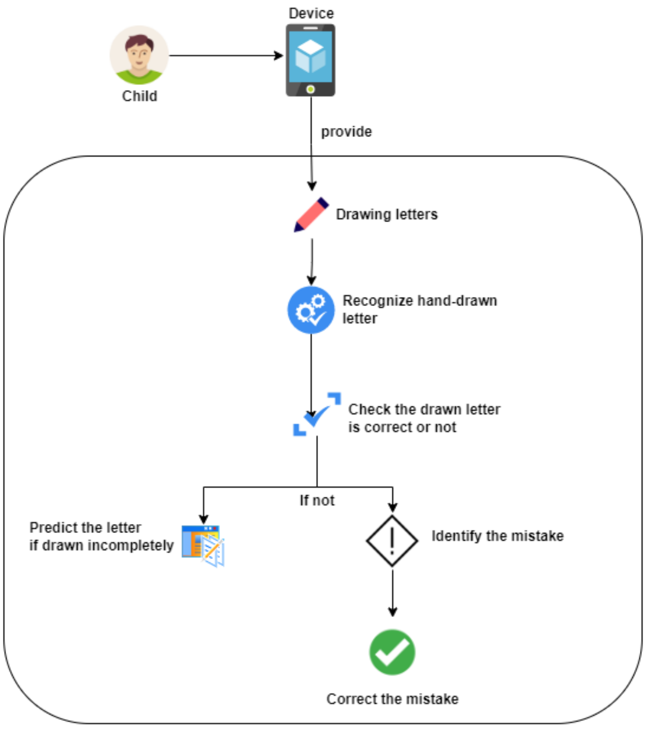

Figure 4.1.2: The overall system diagram for Draw Letters is shown

Figure 4.1.2 illustrates the system architecture of one of the main components of this project, which is related to hand-drawn letters. Furthermore, it has primarily demonstrated how the functions of the hand-drawn alphabet work. This component mainly aimed to identify and predict hand-drawn letters in order to provide suggestions to re-correct the letter in a professional manner.

Focusing on the functionality of the present application, it enables children to freely write letters on a specified area of the screen, recognizes the drawing as a letter in the English alphabet, points out any errors in the fundamental letter drawing, and provides the correct way to draw the letter by itself. Listed below are the steps that user should followed to obtain the final output.

Step 01: Kids need to draw the provided alphabet letters in the given area.

Step 02: After the system automatically detects the hand-drawn letter.

Step 3: If the hand-drawn letter is correct, kids can move on to the next letter.

Step 4: If a hand-drawn letter is wrong, the system automatically identifies the mistake and displays the correct letter in the correct way as a gif video.

The below figure shows how the front-end (UI) side works.

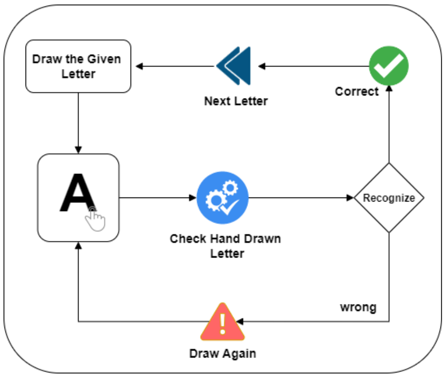

Figure 4.1.3: System Diagram for Frontend

4.2 Designing

When opening the Interactive Learning System for Kids application, first display the splash screen (Figure 4.2.1). On the splash screen, showing some loading effects to get user friendliness. Within a few seconds, the application loads the Home Screen (Figure 4.2.2). After that, users can choose the options that they want to select.

The Figure 4.2.3 illustrates alphabet drawing screen which users can draw the provided letter in the given white area. After clicking the done button. The system automatically predicts the drawn letter and identifies the mistakes, whether they have or not.

If the hand-drawn letter is wrong, it shows a mistake identification screen (Figure 4.2.4). However, identify the mistake from the hand drawn letter. The correct letter is shown as a gif video in the correct way.

If the hand-drawn letter is correct, it shows the Correct Alphabet Identification alert dialog (Figure 4.2.5). After closing the alert dialog, kids can move on to the next letter as provided.

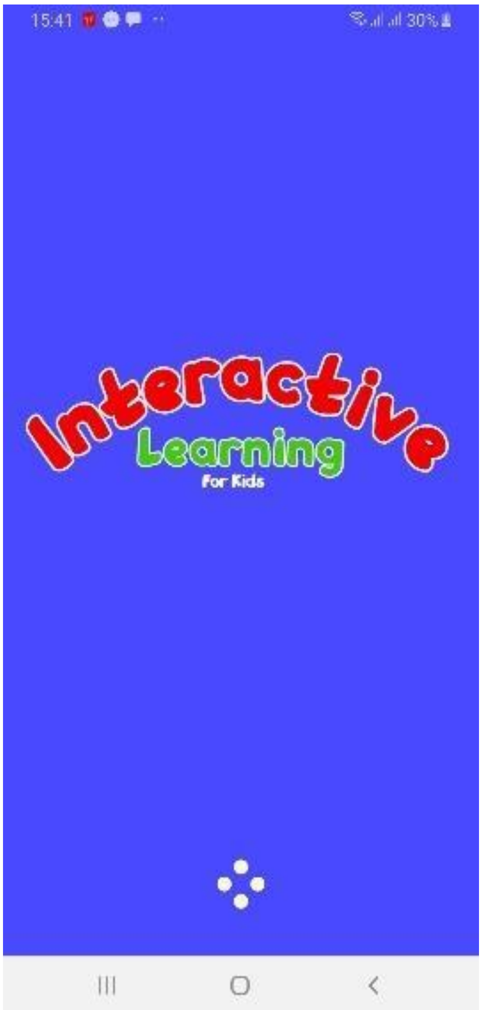

Figure 4.2.1: Splash Screen

Figure 4.2.2: Home Screen

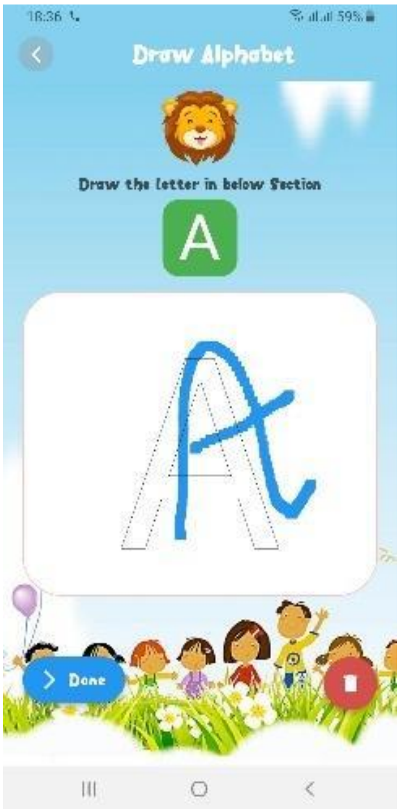

Figure 4.2.4: Mistake Identification

Figure 4.2.3: Alphabet draw screen

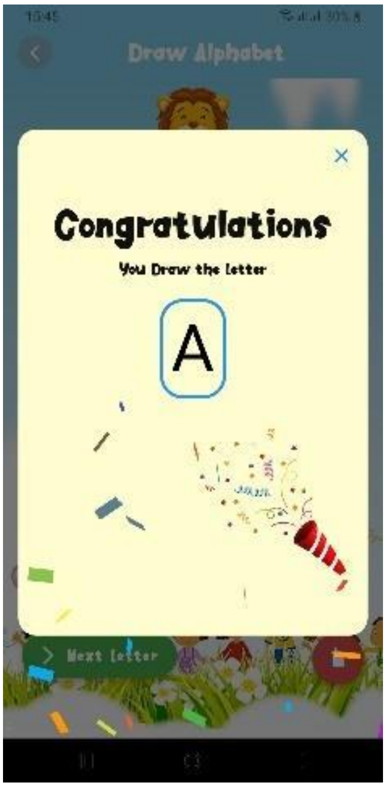

Figure 4.2.5: Correct Alphabet identification

4.3 Model Creation

The study implemented three different models for model creation with the intention of checking the different usage and accuracy results.

- **Model 1:**

The model 1 has used the custom CNN based model with the Kaggle.com dataset. In addition, use three 2D convolution layers with an input shape of 28x28. Moreover Every Con2D layer has a max pooling operation to extract the maximum value from the feature map according to filter size. The reason for using different filters to get the best accuracy result by increasing each Conv2D layer and also ReLU has been used for activation to get a positive result. Softmax activation utilized for multi-class classification problems where required on more than two class labels. Additionally, the Adam optimizer was used to optimize the model. In this model categorical cross entropy is used for the loss function and that is used in multi-class classification tasks.

The below figure shows the model architecture. For the in-depth explanation, Appendix 01 contains the model script.

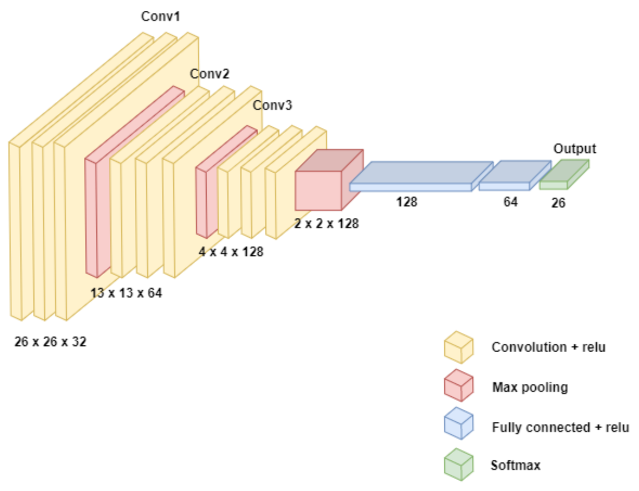

Figure 4.3.1: Model 1 Architecture

- **Model 2:**

The model 02 also Used the same architecture of model 1. But the difference was model 02 changed the input shape as a 224x224.

  - Added the dropout layer after the 2nd Dense layer.
  - Model script shows in appendices 02.

- **Model 3:**

As important and special implementation, model 03 used the transfer learning (VGG16) method as a pretrained model. For this model, the author Added input shape as a 224x224 and global average pooling mode used for feature extraction. Furthermore added new layers to the model as a custom way which means cut off the middle layer in the original VGG16 model layers. Then added 5 new layers (1 Flatten layer & 4 Dense layers). Model 3 utilized Relu & Softmax for the activation, and used categorical cross entropy for the loss function. In addition, researchers used the Adam optimizer to optimize the model.

4.4 Software Solution

The agile approach will be used to analyze the software development life cycle. And in the agile methodology, Scrum will be the methodology that will be used. Scrum is a simple way to execute agile. Scrum is currently the most common development methodology, not only for software but also for finance and analysis.In addition, authors used MS-teams and WhatsApp to gather the meetings and share the necessary information related to the research works.

4.5 Requirement Gathering and Analysis

As a requirement for this study, data was collected from nursery teachers, parents, and children. The collected data contained images, books, experiences, practical instructions, etc.

- **User Requirements**
    - User-friendliness.
    - Provide solutions to improve knowledge of writing letters.
    - Facilitate quick response.

- **Functional Requirements**
    - Recognize hand-drawn letters.
    - Predict the letter if the student draws the letter incompletely.
    - Identify mistakes made by the student when drawing the letter and suggest solutions.

- **Data Set**
    - The models were primarily built using two data sets. One dataset was obtained from kaggle.com, and another from the reference's person.

01) Old Dataset

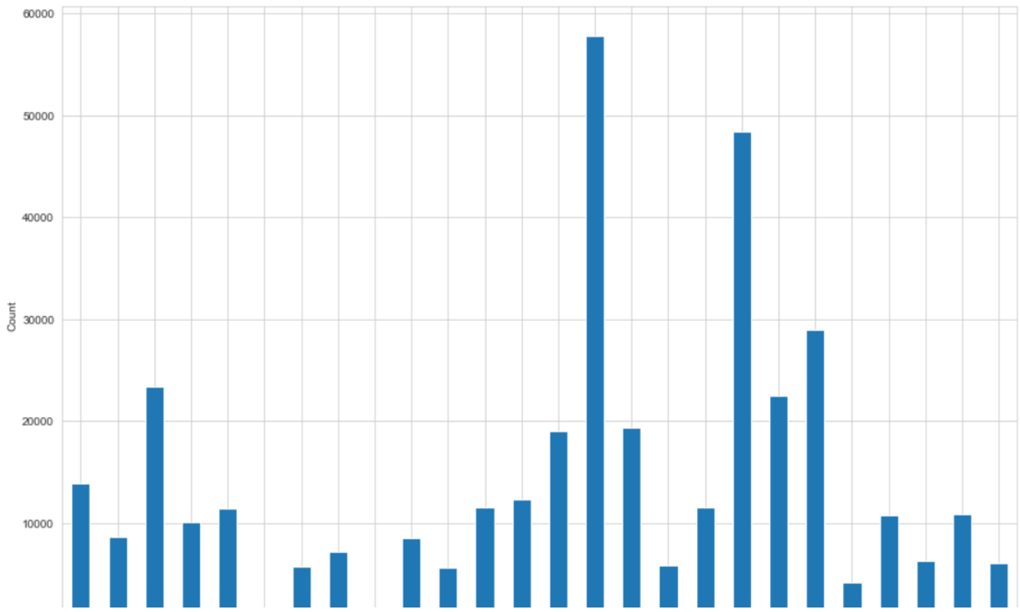

Figure 4.5.1: Kaggle.com dataset

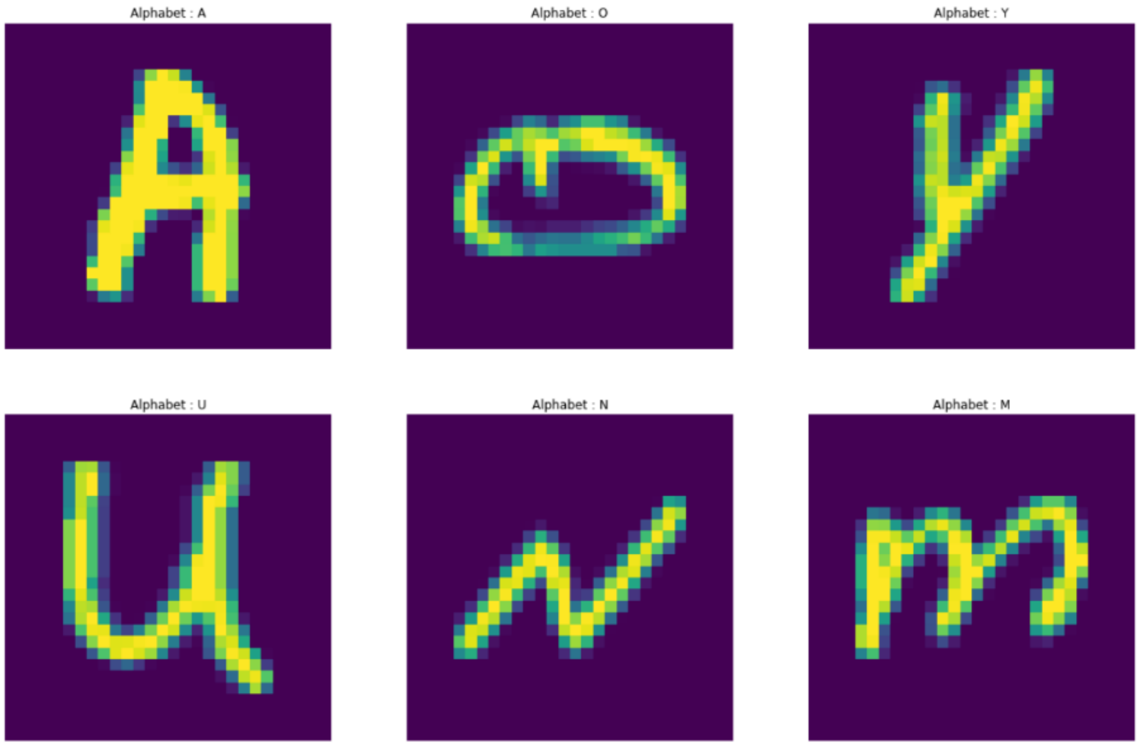

Figure 4.5.2: Training data

This dataset contains 26 folders (A-Z) containing handwritten images in size 28**28 pixels.*It contains 370000+ images in data set. Then all the images are converted into the CSV format and also images are taken from NIST(https://www.nist.gov/srd/nist-special-database-19) NMIST large dataset and few other sources which were then formatted as mentioned above.

02) New Dataset

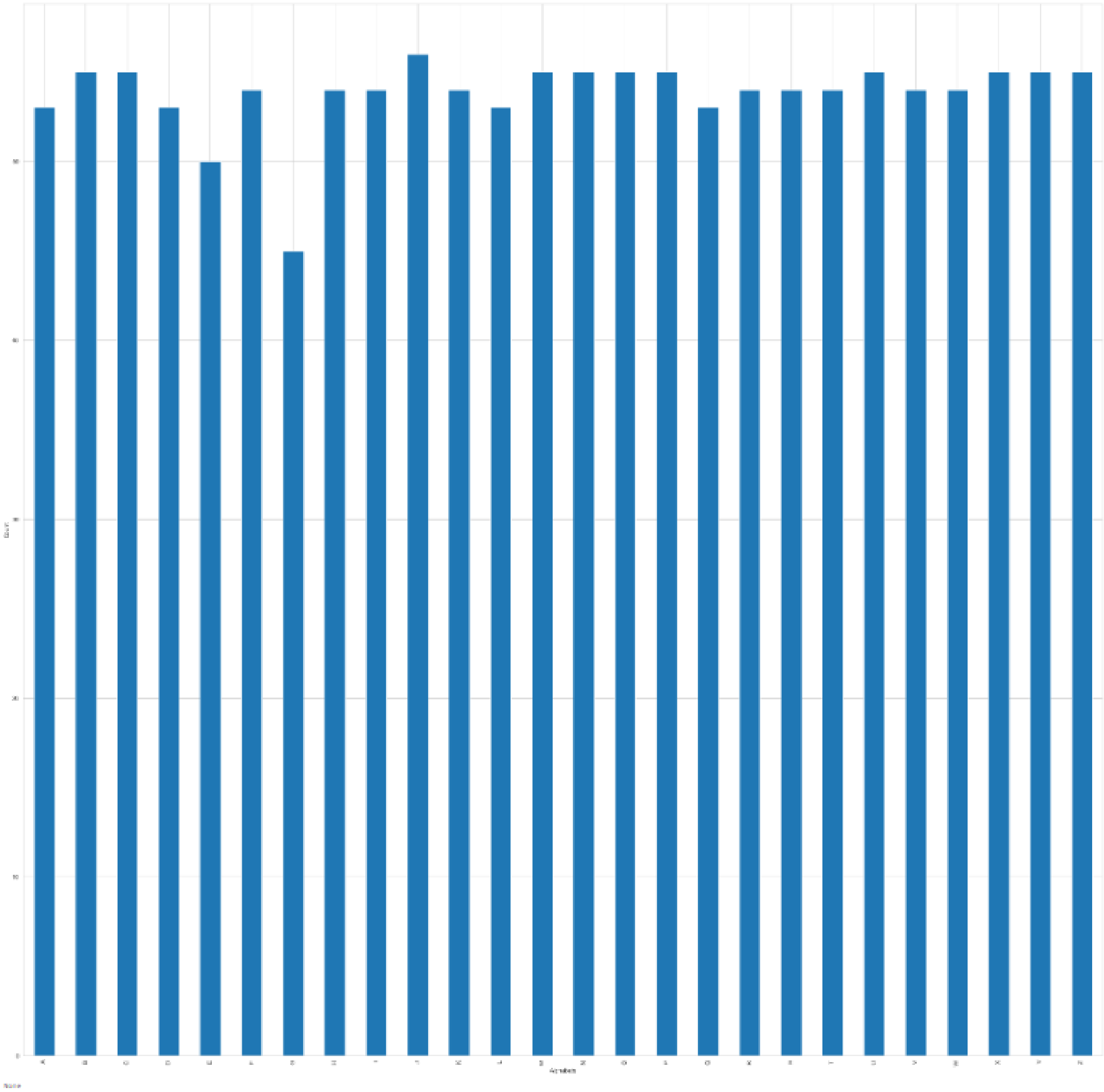

Figure 4.5.3: New dataset

Figure 4.5.4: Training data

This dataset was gathered from an external resource person. He got this dataset from stpete_ishii Engineer at TZ TECHNO, Kobe, Hyogo, Japan. Previously, this data set included different kinds of images. such as numbers, alphabets, and symbols. An external resource person regenerated this dataset as particular alphabet letters.The dataset contains 26 labels (A-Z) containing handwritten images of size *224*224* pixels.Morevere, It has contains 1400+ images in its data set.All the images are converted into the CSV format. In Particular, The present study created a new python script to convert the image to the CSV format to convert it in the CSV format.

4.6 Commercialization

To commercialize the app, authors expect to utilize several types of effective marketing strategies. Authors will use the following strategies to promote the application: creating a website, social media marketing, creating a teaser video, considering alternative app stores, creating an eye-catching app icon, and naming. Furthermore, Authors expected to provide a user subscription plan & partnerships to commercialize the application. Example for Subscription plan is shown in Figure 4.6.1

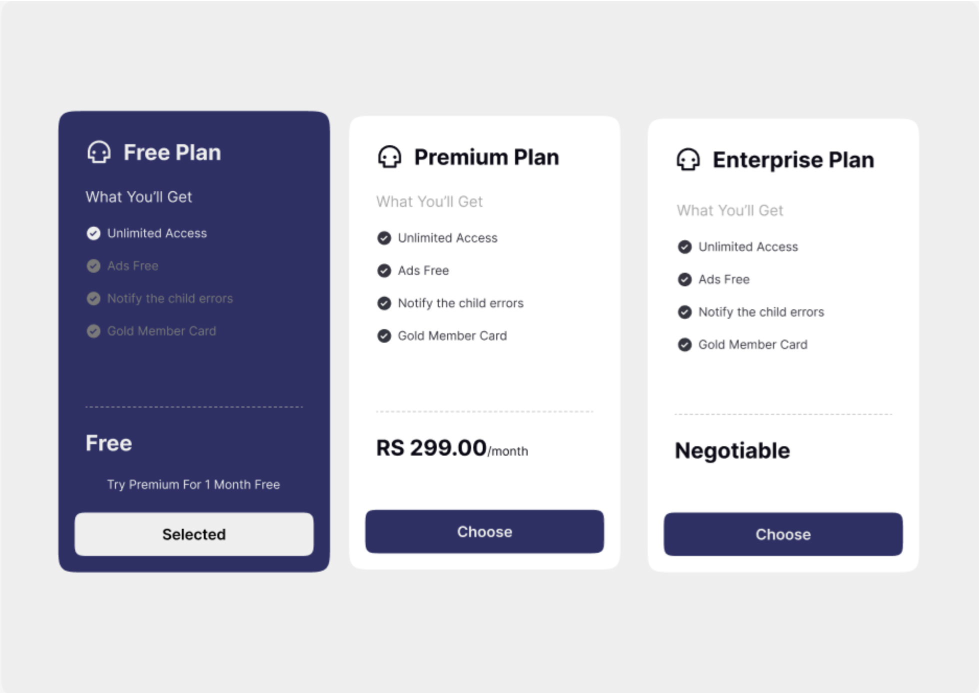

Figure 4.6.1: Example Subscription plan

Application creators hope to create and design a website that is fully functional and mobile-friendly. That will allow them to attract more customers for the product as well as distribute details of the product to national and international markets. Aligning with that, developers intend to create an eye-catching app icon and an attractive name for the application that is suitable for promoting in the market. It is really important to select the right social media platforms to promote the mobile application among the number of social media channels available on the market.

The selection of the platform should be more focused on the ability to reach a targeted audience. In order to create an app, creators hope to implement some Facebook and Instagram advertisements. For posting product details, we propose to create a teaser video as a simple 30 second commercial by describing the why, how, and what details of the application. Furthermore, it is important to connect with social media groups among parents and teachers that are related to pre-school education. Furthermore, the creators intend to consider alternative app stores other than Google Play and Apple Store, such as GetJar, Amazon Appstore, and Opera Mobile Store, in addition to Google Play and Apple Store.

5. IMPLEMENTATION

- **Mobile Application**

  The outcome of this paper is a mobile application that enhances the ability to write letters for primary kids. The main purpose of the mobile application is to provide knowledge of alphabet letters for primary kids.

- **Languages**

  The study focused on deep learning with transfer learning (VGG16) using flutter to build this mobile application. Furthermore, deep learning and transfer learning (VGG16) were used to train the model and the flutter was used to design the applications.

- **Image to CSV converter**

  The dataset had to be changed to a CSV format. The dataset had to be changed to a CSV format. The study used the method of writing a Python script to convert the images into CSV format. Author followed the YouTube tutorials and reference websites to create this Python script.

- **Model Script**

  The model was created using a Python-based script and also implemented the three models with two data sets: CNN-based and Transfer Learning (VGG16).

  - Model 1: Used the Kaggle dataset and created the custom model.
  - Model 2: Used the new dataset and created the custom model.
  - Model 3: Used the new dataset and created the transfer learning (VGG16) model.

- **Mobile Application Connect with the Model**

  To connect the model and the mobile application, created model save as a tflite file (A Flutter plugin for accessing the TensorFlow Lite API). This plugin supports image classification, object detection (SSD and YOLO), Pix2Pix, Deeplab, and PoseNet on both iOS and Android).

  This Tflite file moves on to the flutter project (for the asset folder). In the flutter project, load the TFlite model to get the model information and also pass the inputs as a pixel into the model to get the output. However, this process worked in real time to get more user-attractiveness and friendliness.

- **Mistake Identification**

  The image_comapre flutter library was used for error detection. In the image_compare library, there are different kinds of algorithms. But in the current case, the author used the IMED algorithm (IMage Euclidean Distance). This IMED algorithm is a pixel comparison algorithm.

  In the IMED, images are resized to the same dimensions (if dimensions don't match) and are grayscale. A gaussian blur was applied when calculating the distance between pixel intensities. Spatial relationships were considered within the gaussian function to reduce the effect of minor perturbations.

  The outcome of the IMED algorithm, sum of image euclidean distances between each pixel (RGB value), bounded by the maximum distance possible given two images (hand-drawn alphabet images and original alphabet images).

6. RESULTS & DISCUSSION

This section discusses the results and findings of the prediction of the alphabet letters and identifies the mistakes of the hand-drawn alphabet letters.

6.1 Results

Using CNN & Transfer Learning (VGG16) based model, a balanced 2 dataset containing 370000+ and 1400+ images have been selected for the prediction and identify the alphabet letters. To identify the mistakes used the image_compare library. However, using 3 models got 3 different accuracy results. But whatever calculation, the main purpose is to enhance the Kids' handwritten skills and knowledge.

According to model 1 accuracy got a 98+ accuracy result. Figure 6.1.1 shows how the test & train accuracy results increase. Using 5 epochs could be able to reach get good accuracy results.

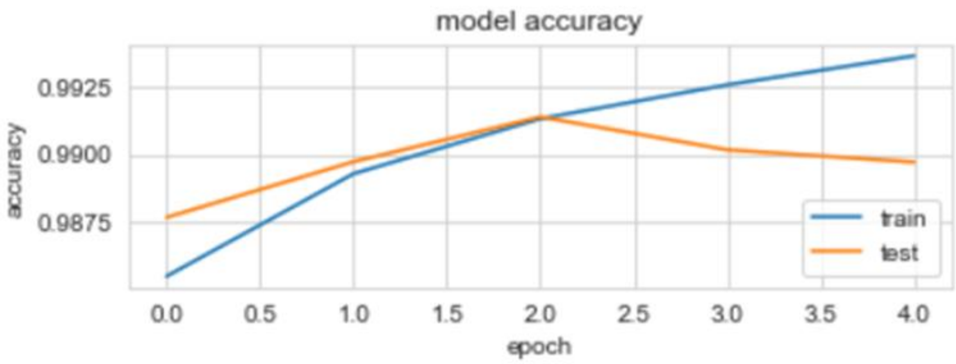

Figure 6.1.1: Model 1 Accuracy Result

In the 2 Model changed the dataset from Kaggle.com (Big) to a new dataset (Small). After doing that study I received a more than 75% accuracy result. Figure 6.1.2 shows the accuracy result of model 2.

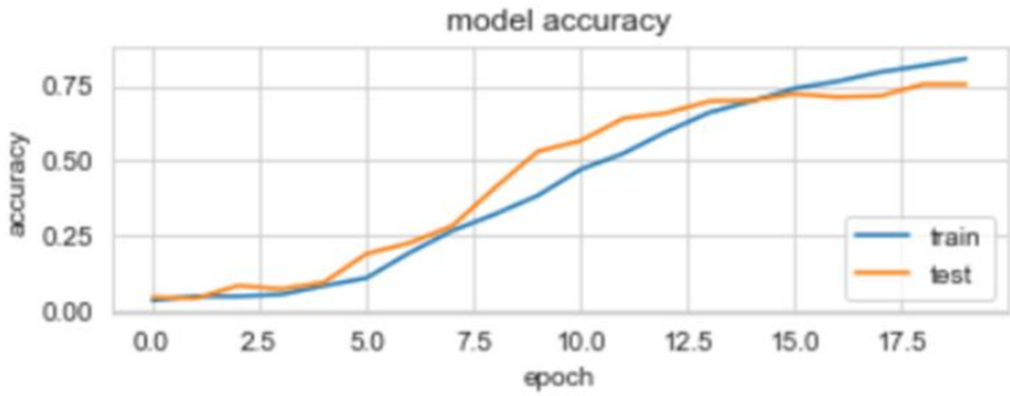

Figure 6.1.2: Model 2 Accuracy Result

Using VGG16, the study created the third model and received a more than 45% accuracy result. The reason behind the low accuracy rate was the small dataset availability in the new data set as a pre-trained model. The figure 6.1.3 shows the test & train accuracy results parallelly.

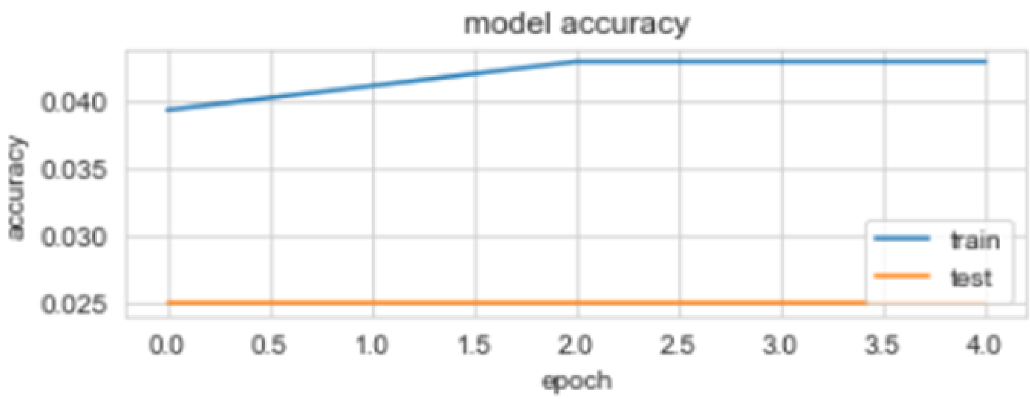

Figure 6.1.3: Model 2 test and train accuracy result

After comparing each model, the author concluded and choosed model 1 as the best accuracy model.

Prediction of Alphabet Letter: Since all the alphabet letters predict the correct alphabet letter as much as possible when hand drawing the letters.

Mistake Identify: Using the image_compare Algorithm, could be able to identify the mistakes with differentiation percentage.

Mistake Correction: Shows a gif video of the alphabet letter on the screen in the correct way.

6.2 Research Findings

After conducting the data analysis described in the preceding figure 6.1, author, came to the conclusion that only a few of the features have a direct influence when it comes to recognizing and predicting the hand drawn letters as well as locating the errors. Following the completion of the analysis, the following list presents the selected features.

- Ability to Erase hand drawn letters
- Ability to draw alphabet freely
- Ability to quick response

6.3 Discussion

Most importantly, letters that have been drawn by hand can be recognized by this application. Therefore, in order to use the application effectively, users will need to draw the alphabet letters on the screen that has been provided to them. In the beginning, it's possible that children will need some assistance from an adult in order to become familiar with the mobile application. Following table 6.3.1 has compared the current research findings in relation with what already implemented in the subject area. When compared to prior studies, the authors of the present work discovered that it added more comprehensive work by including mistakes identifications, predictions, and efforts to correct those mistakes with correct way.

Table 6.3.1: Comparison of the Research works

Research	Drawing Pattern	Mistake Identify	Predication	Correction
Neural Networks for Handwritten English Alphabet Recognition	Image drawing	✗	✓	✗
Development of an Automated Handwriting Analysis System	Image drawing	✗	✓	✗
Graphology for Farsi Handwriting Using Image Processing Techniques	Image drawing	✗	✓	✗
Augmented Reality and Machine Learning-Based Mobile App to Learn Writing	Mobile App Screen with given image	✗	✗	✗
Interactive Learning System for kids (Present Application)	Mobile App Screen with free draw	✓	✓	✓

7. CONCLUSIONS

In this study authors have presented a novel approach for children, which is to improve their knowledge and ability of the alphabet while also helping them improve their ability to correctly draw it by hand. The tool that researchers developed to make predictions and identify errors can also be used to improve children's knowledge of the alphabet, as was covered in the methodology section. Any android device could have this Interactive system's Mobile Application downloaded and installed on it because it was developed as a Mobile Application.

8. REFERENCE LIST

[1] L. D. Harmon, "Automatic recognition of print and script," in Proceedings of the IEEE, vol. 60, no. 10, pp. 1165-1176, Oct. 1972, doi: 10.1109/PROC.1972.8878.

[2] Izzat Fauzi Bin Hamidi (2020) Development of Augmented Reality App for Kids to Learn Basic Jawi Alphabet. DOI: 10.13140/RG.2.2.14740.24969

[3] Yusuf Perwej, Ashish Chaturvedi, Neural Networks for Handwritten English Alphabet Recognition International Journal of Computer Applications (0975 – 8887) Volume 20– No.7, April 2011

[4] Ammar H. Safar, Ali A. Al-Jafar, Zainab H. Al-Yousefi, The Effectiveness of Using Augmented Reality Apps in Teaching the English Alphabet to Kindergarten Children: A Case Study in the State of Kuwait EURASIA Journal of Mathematics Science and Technology Education ISSN 1305-8223 (online) 1305-8215 (print) 2017 13(2):417-440

[5] Vikram Kamath, Nikhil Ramaswamy, P. Navin Karanth, Vijay Desai and S. M. Kulkarni DEVELOPMENT OF AN AUTOMATED HANDWRITING ANALYSIS SYSTEM, ARPN Journal of Engineering and Applied Sciences VOL. 6, NO. 9, SEPTEMBER 2011, ISSN 1819-6608

[6] Aizan, N.L., Mansor, E.I., & Mahmod, R. (2014). Preschool children handwriting evaluation on paper-based and tablet-based settings.

[7] Géraldine Jordan, Fanny Michaud & Marie-Laure Kaiser (2016) Effectiveness of an intensive handwriting program for first grade students using the application LetterSchool: A pilot study, Journal of Occupational Therapy, Schools, & Early Intervention, 9:2, 176-184, DOI: 10.1080/19411243.2016.1178034

[8] A. K. S. Alvaro, R. L. D. J. Dela Cruz, D. M. T. Fonseca and M. J. C. Samonte, "Basic handwriting instructor for kids using OCR as an Evaluator," 2010 International Conference on

Networking and Information Technology, 2010, pp. 265-268, doi: 10.1109/ICNIT.2010.5508516.

[9] B. R. Maxim, N. V. Patel, N. D. Martineau, and M. Schwartz, “Work in progress - learning via gaming: An immersive environment for teaching kids handwriting,” 2007 37th annual frontiers in education conference - global engineering: knowledge without borders, opportunities without passports, vol. 2007, pp. T1B-3-T1B-4, Oct. 2007.

[10] Loey, Mohamed & Elsawy, Ahmed & El-Bakry, Hazem. (2016). Arab Kids Tutor (AKT) System For Handwriting Stroke Errors Detection. INTERNATIONAL JOURNAL OF TECHNOLOGY ENHANCEMENTS AND EMERGING ENGINEERING RESEARCH. 4. 42-49.

[11] Hassanin Al-Barhamtoshy, Sherif Abdou, Fakhraddin A. Al-Wajih (2012). A Toolkit for Teaching Arabic Handwriting. International Journal of Computer Applications (0975 – 8887) Volume 49

[12] Steven Sybenga & Yves Rybarczyk (2018) Using Machine Learning and Image Processing for Character Recognition: An Application for Teaching Handwriting

[13] Yanikoglu, B., Gogus, A., & Inal, E. (2017). Use of handwriting recognition technologies in tablet-based learning modules for first grade education. Educational Technology Research and Development, 65(5), 1369–1388.

[14] Iwayama, N., Akiyama, K., Tanaka, H., Tamura, H., & Ishigaki, K. (2004). Handwriting-based learning materials on a tablet PC: a prototype and its practical studies in an elementary school. Ninth International Workshop on Frontiers in Handwriting Recognition, 533-538.

[15] Hu, Z., Xu, Y., Huang, L., & Leung, H. (2009). A Chinese Handwriting Education System with Automatic Error Detection. *J. Softw., 4*, 101-107.

[16] Kristi Hape OTD, MHS, OTR, Nicole Flood MOT, OTR, Kari McArthur MOT, OTR, Chantel Sidara MOT, OTR, Christina Stephens MOT, OTR & Kayla Welsh MOT, OTR (2014) A Pilot Study of the Effectiveness of the Handwriting Without Tears® Curriculum in First Grade, Journal of Occupational Therapy, Schools, & Early Intervention, 7:3-4, 284-293,

9. APPENDICES

```
# creating the model

model = Sequential()
model.add(Conv2D(filters=32, kernel_size=(3, 3),
          activation='relu', input_shape=(28, 28, 1)))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))

model.add(Conv2D(filters=64, kernel_size=(3, 3),
          activation='relu', padding='same'))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))

model.add(Conv2D(filters=128, kernel_size=(3, 3),
          activation='relu', padding='valid'))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))

model.add(Flatten())  # output

model.add(Dense(128, activation="relu"))
model.add(Dense(64, activation="relu"))
# output layer
model.add(Dense(26, activation="softmax"))
# compile
model.compile(optimizer=adam_v2.Adam(learning_rate=0.001),
              loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
```

Appendices 01 : Model 1

```python
# model
model = Sequential()
model.add(Conv2D(filters=32, kernel_size=(3, 3),
          activation='relu', input_shape=(224, 224, 1)))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))

model.add(Conv2D(filters=64, kernel_size=(3, 3),
          activation='relu', padding='same'))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))

model.add(Conv2D(filters=128, kernel_size=(3, 3),
          activation='relu', padding='valid'))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))

model.add(Flatten())  # output

model.add(Dense(128, activation="relu"))
model.add(Dense(64, activation="relu"))
model.add(Dropout(0.35))
# output layer
model.add(Dense(26, activation="softmax"))
# compile
model.compile(optimizer=adam_v2.Adam(learning_rate=0.001),
              loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
```

Appendices 02: Model 2

```python
# model create
base_model = VGG16(
    weights=None,
    include_top=True,
    input_shape=(224, 224, 1),
    pooling='avg'
)
for layer in base_model.layers:
    layer.trainable = False

x = layers.Flatten()(base_model.output)
x = layers.Dense(512, activation="relu")(x)
x = layers.Dense(128, activation="relu")(x)
x = layers.Dense(64, activation="relu")(x)
x = layers.Dense(26, activation="softmax")(x)
model = Model(base_model.input,x)

model.compile(loss='categorical_crossentropy', optimizer=adam_v2.Adam(learning_rate=0.001), metrics=['accuracy'])
model.summary()
```

Appendices 03: Model 3 (VGG16)

```python
def createFileList(myDir, format='.png'):
    fileList = []
    print(myDir)
    for root, dirs, files in os.walk(myDir, topdown=False):
        for name in files:
            if name.endswith(format):
                fullName = os.path.join(root, name)
                fileList.append(fullName)
    return fileList
```

Appendices 04: File Creating for images (Image to CSV python script)

```python
# load the original image
myFileList = createFileList(
    '../dataset/en/endata/alpabet_with_folder/224x224/25/')

for file in myFileList:
    print(file)
    img_file = Image.open(file)
    print(img_file.size)

    # get original image parameters...
    width, height = img_file.size
    format = img_file.format
    mode = img_file.mode

    # Make image Greyscale
    img_grey = img_file.convert('L')

    # Save Greyscale values
    value = np.asarray(img_grey.getdata(), dtype=np.int64).reshape(
        (img_grey.size[1], img_grey.size[0]))
    value = value.flatten()
    df = pd.DataFrame(value).T
    df.insert(loc=0, value="25", column="A")
    print(df.head())
    print(value)
    with open("../dataset/en/endata/alpahbet_csv/224x224_trans.csv", 'a') as f:
        df.to_csv(f, index=False, header=False, mode='a')
```

Appendices 05: convert images to csv format (Image to CSV python script)

```
compareImagesWithBytes(List<Offset> points, String alphbetLetter) async{

  print("Letter >>>>>> $alphbetLetter");
  final ByteData bytes = await rootBundle.load('assets/images/letters/$alphbetLetter.png');
  Uint8List bytes1 = bytes.buffer.asUint8List();

  final picture = toPicture(points);
  final image = await picture.toImage(mnistSize, mnistSize);
  ByteData? imgBytes = await image.toByteData(format: ImageByteFormat.png);
  Uint8List? bytes2 = imgBytes?.buffer.asUint8List();

  print(bytes1);
  print(bytes2);

  try{
    /// Calculate pixel Matching difference between two bytes of images
    var byteResult = await compareImages(src1: bytes1, src2: bytes2,algorithm:IMED(blurRatio: 0.05));
    int diff = (byteResult * 100).toInt();
    print("Difference : $diff%");

    return diff;
  }catch(e){
    print(e);
  }
}
```

Appendices 06: Mistake Identification

```
Picture toPicture(List<Offset> points){

  final _whitePaint = Paint()
      ..strokeCap = StrokeCap.round
      ..color = kWhite
      ..strokeWidth = kStrokeWidth;

  final _bgPaint = Paint()..color = kBrushBlack;
  final _canvasCullRect = Rect.fromPoints(Offset(0,0),Offset(mnistSize.toDouble(),mnistSize.toDouble()));
  final recorder = PictureRecorder();
  final canvas = Canvas(recorder,_canvasCullRect)..scale(mnistSize/canvasSize);

  canvas.drawRect(Rect.fromLTWH(0, 0, 36, 36), _bgPaint);

  for(int i = 0; i < points.length - 1; i++){
    if(points[i] != null && points[i + 1] != null){
      canvas.drawLine(points[i], points[i + 1], _whitePaint);
    }
  }
  return recorder.endRecording();
}
```

Appendices 07: Bytes convert to Image

```
Paint drawingArea = Paint()
  ..strokeCap = StrokeCap.square
  ..isAntiAlias = true
  ..color = kBrushColor
  ..strokeWidth = kStrokeWidth;

class DrawAlphabet extends CustomPainter{
  DrawAlphabet({required this.offsetPoints});
  List<Offset> offsetPoints;
  final path = Path();

  @override
  void paint(Canvas canvas, Size size) {
    for(int i = 0; i < offsetPoints.length - 1; i++){
      if(offsetPoints[i] != null && offsetPoints[i+1] != null){
        canvas.drawLine(offsetPoints[i], offsetPoints[i], drawingArea);
      }
    }
  }

  @override
  bool shouldRepaint(covariant CustomPainter oldDelegate) => true;

}
```

Appendices 08: Custom Painter

```
Future<String> getPrediction(Uint8List imgAsList) async{

  final resultBytes = List.filled(mnistSize * mnistSize,0.0);

  int index = 0;
  for (int i = 0; i < imgAsList.lengthInBytes; i += 4) {
    final r = imgAsList[i];
    final g = imgAsList[i+1];
    final b = imgAsList[i+2];

    // the mean of R,G,B channel into single GrayScale
    resultBytes[index] = ((r + g + b) / 3.0) / 255.0;
    index++;
  }

  var input = resultBytes.reshape([1, 28, 28, 1]);
  var output = List.filled(1*26, null, growable: false).reshape([1, 26]);

  // used to set GPUDelegate
  InterpreterOptions interpreterOptions = InterpreterOptions();

  // Track how long it took to do inference
  int startTime = DateTime.now().millisecondsSinceEpoch;

  try {
    //load the model class
    Interpreter interpreter = await Interpreter.fromAsset("mnist.tflite", options: interpreterOptions);
    interpreter.run(input, output);
  } catch (e) {
    print('Error loading or running model: ' + e.toString());
  }

  int endTime = DateTime.now().millisecondsSinceEpoch;

  // Obtain the highest alphabet letter from the output of the model
  double highestProb = 0.0;
  String digitPred = "";

  for (int i = 0; i < output[0]!.length; i++) {
    if (output[0]![i] > highestProb) {
      highestProb = output[0]![i];
      digitPred = i.toString();
    }
  }
  print(digitPred);
  return digitPred;
```

Appendices 09: Alphabet Prediction