# INTERACTIVE LEARNING SYSTEM FOR KIDS

2022-254

# TEAM

**Supervisor**
**Ms. Namalie Walgampaya**

**Co-Supervisor**
**Ms. Hansi De Silva**

Dhanushka Dilshan
IT19138732

Shehan Dhanuddara
IT19167206

Janitha Tharaka
IT19148014

Nisal Ravindu
IT19197838

# CONTENT



# INTRODUCTION

- The education field has been revolutionized by introduction of information technology-based tools.
- The effective integration of educational technology and early childhood education requirements are limited.
- The primary aim is to develop an interactive learning platform that provides teaching aids to primary level students without the need of manual supervision.
- Researchers integrated advanced technology and key primary level activities
    - Letter based activities
    - Image based activities
    - Shapes based activities
    - Color based activities

# OBJECTIVES

## ❑ Main Objective

- To create an interactive learning system to enhance knowledge and skills of primary level students .

## ❑ Sub Objective

- To identify and predict hand-drawn letters, in order to provide suggestions to recorrect the letter in professional manner.
- To recognize shapes and number of shapes by art drawn ,according to the instructions given by system.
- To identify color of drawing object in order to provide corrections and suggestions for the given draw
- Extract features from given image & generate caption depending on that extracted image features

# IT19138732|M.G.D.D.B. Ekanayaka

**B.Sc. (Hons) Degree in Information Technology Specialized in Software Engineering**

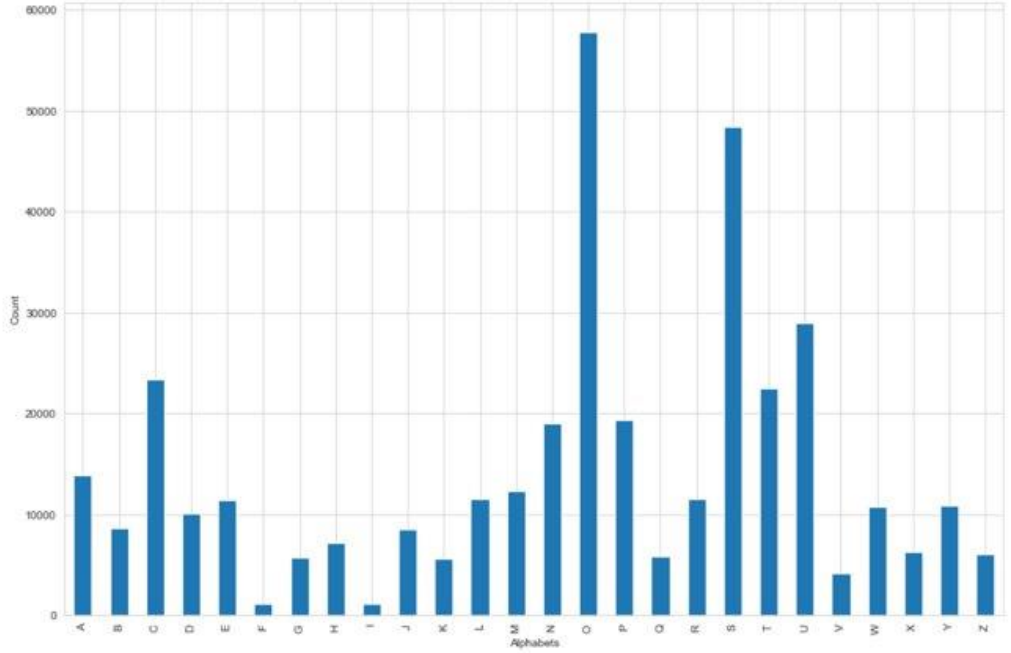
# Introduction - Interactive learning system of English alphabet for kids

- Focusing on children's handwriting ability of English alphabet.
- Primary aims - To identify and predict hand-drawn letters, in order to provide suggestions to recorrect .
- Existing applications - Allowed children to draw a letter on a provided letter-shaped image (image drawing).
- Present research work
    - Allowing children to write letters on given screen area.
    - predicting the drawing as English letter.
    - identifying mistakes in the basic letter drawing.
    - Demonstrating correct way to draw the letter by the application.

- This system would be act as self-learning platform to children to improve alphabetical knowledge and skills.

# Implementation..

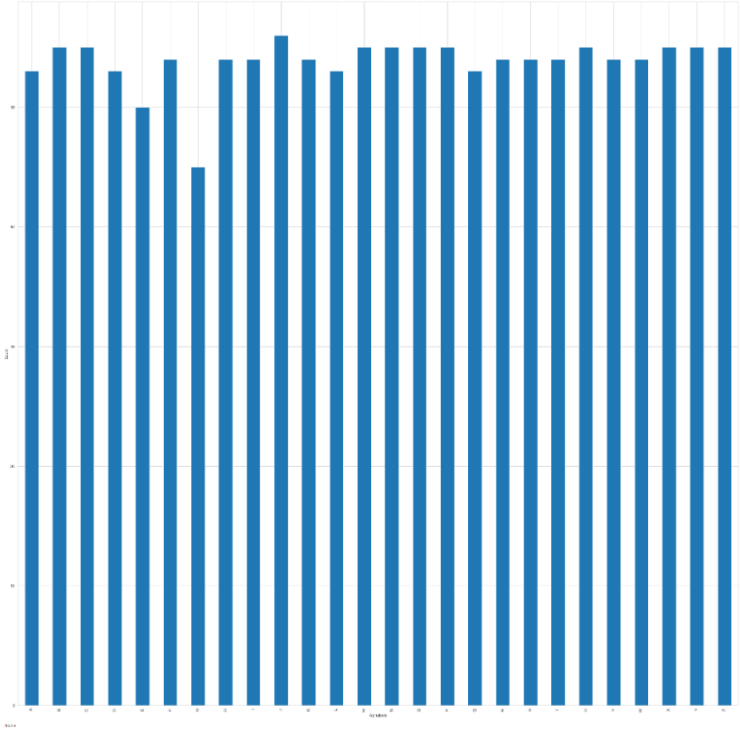
- Dataset – Old




- The dataset contains 26 folders (A-Z) containing handwritten images in size 28**28 pixels.*
- Contain 370000+ images in data set.
- All the images are converted into the CSV format.

# Implementation..

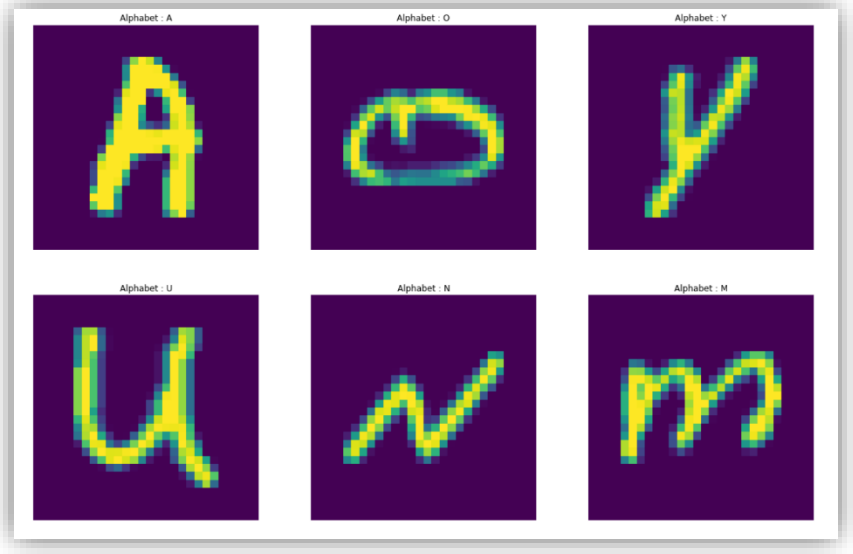
- Dataset -New



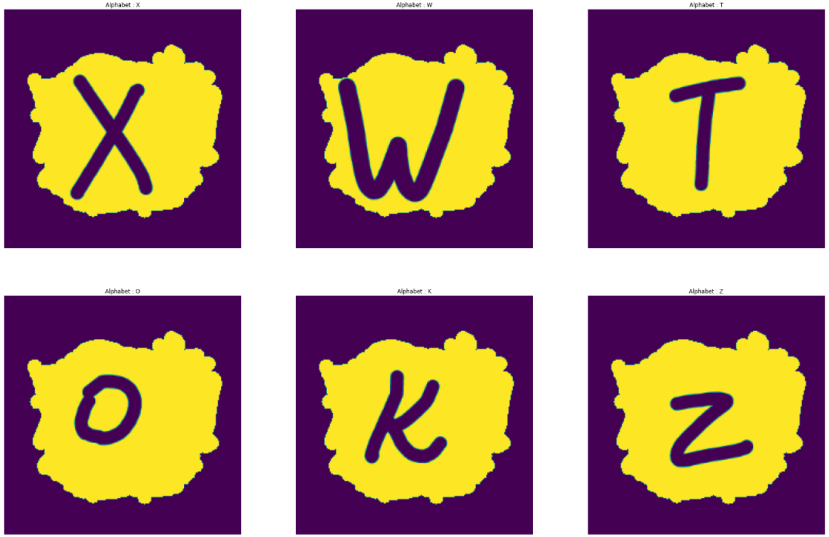
- The dataset contains 26 folders (A-Z) containing handwritten images in size *224*224* pixels.
- Contain 1400+ images in data set.
- All the images are converted into the CSV format.
- To convert the CSV format, create the new python code to convert the image to CSV format

# Implementation..

- Old & New Training data



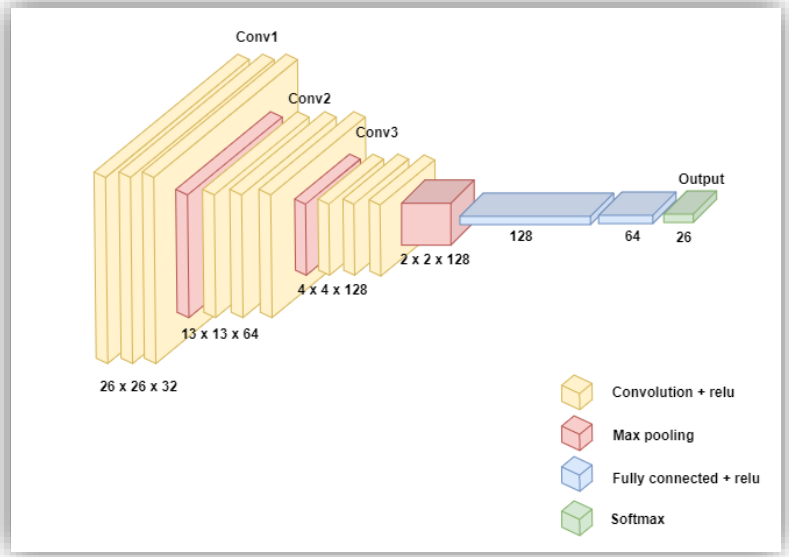

*Old Training data*




*New Training data*

# Implementation..

- Model Creation – Old Data-Set




- Use three 2D Convolution layers.
- Every Con2D layers have max pooling operation to extract maximum value from the Feature map according to filter size.

# Implementation..

## ❑ Model Creation - Old Data-Set

```
# 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
model.summary()
```

- Used different filters to get the best accuracy result by increasing each Conv2D layers.
- Used ReLU for activation to get the positive result.
- Used Softmax activation for multi-class classification problems where required on more than two class labels.

# Implementation..

## ❑ Model Creation - Old Data-Set

```
# 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
model.summary()
```

- Adam optimizer used for the optimize the model.
- Used Categorical cross entropy for the loss function and that is used in multi-class classification tasks.

# Implementation..

## ❑ Model Creation – New Data-Set

```
# 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()
```

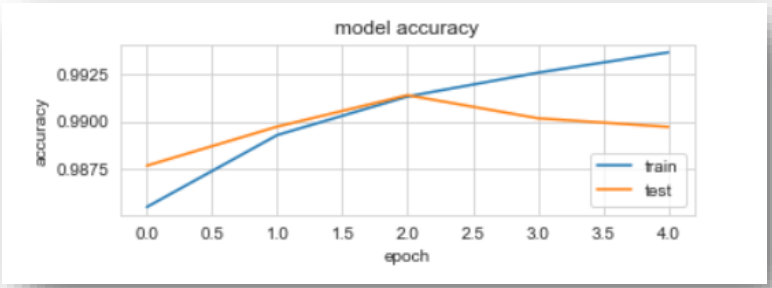
- To create this model only changed the input shape size as (224,224,1) cause the image size is 224x224 px.

# Implementation..

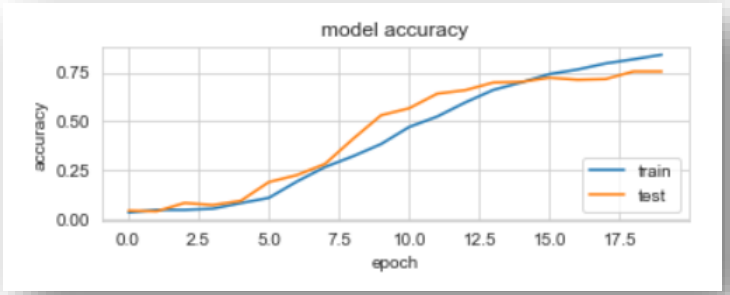
- Model Creation – New Data-Set with Transfer Learning (VGG16)

```
# 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()
```

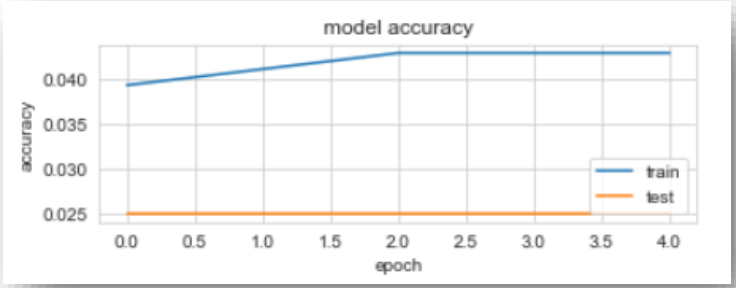
# Results




**Old model accuracy – 98%+**

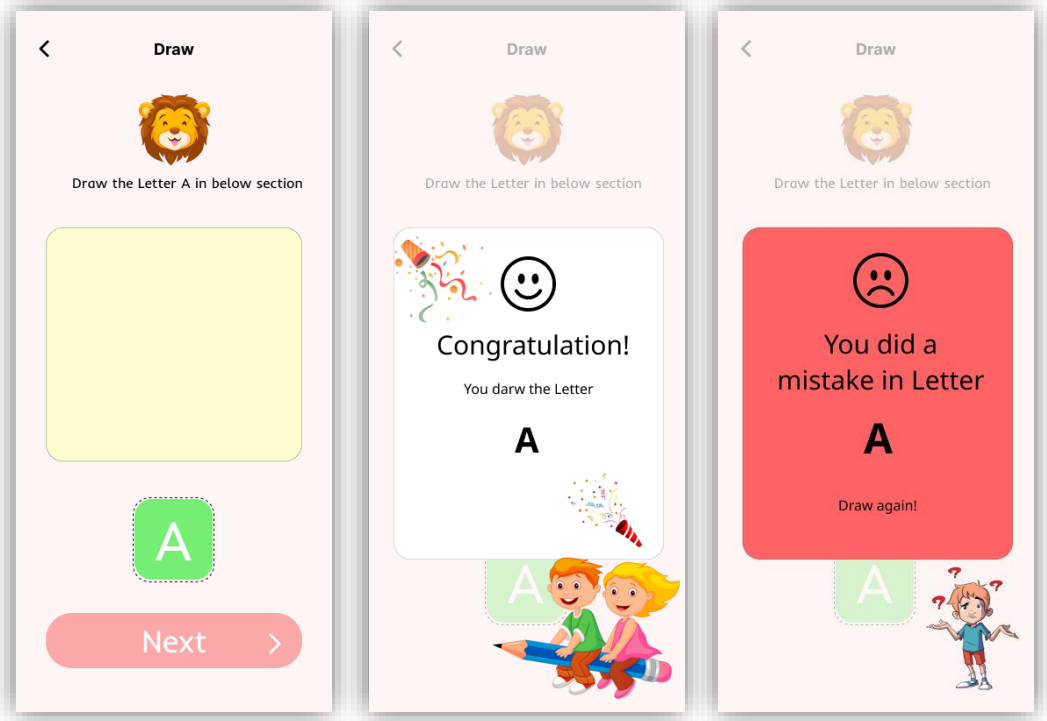



**New model accuracy – 75%+**



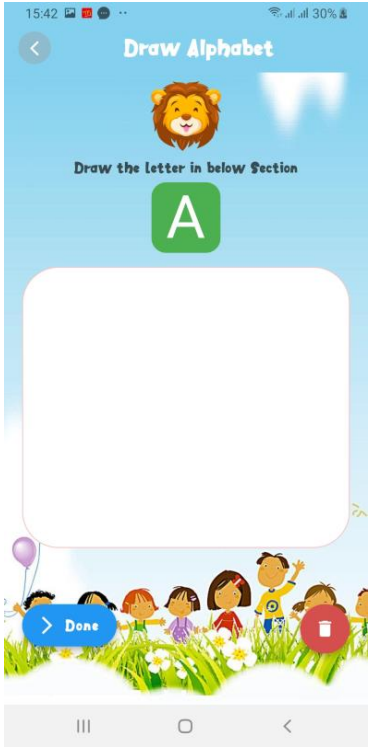

**VGG16 model accuracy – 45%+**

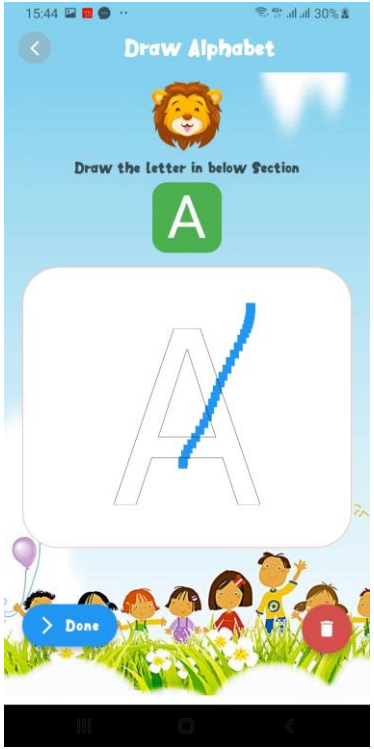
# Designing




- Use Figma tool to design the Mobile Interface.
- Here shows pre-design interfaces.

# Designing..

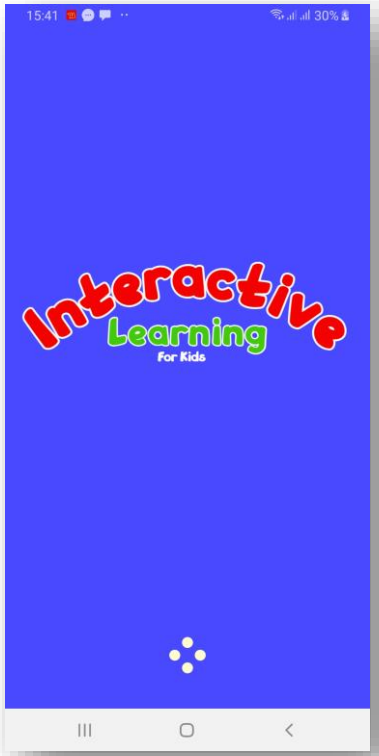




- This images shows the actual UI output of the application.

# Designing..

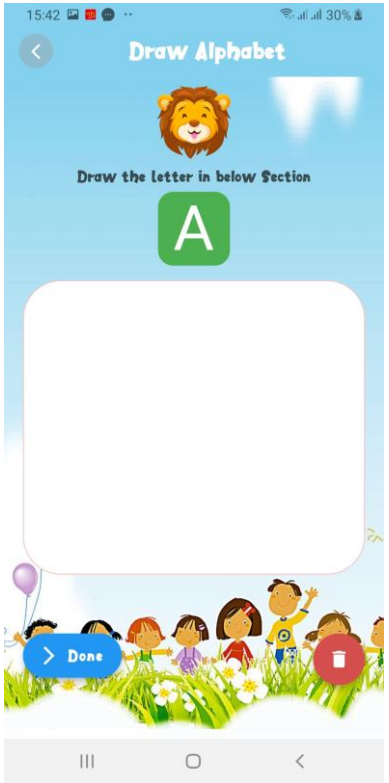



- This is the Splash Screen in the Mobile application for our research.

# Designing..


- This is the Actual Home Screen in the Mobile application for our research.

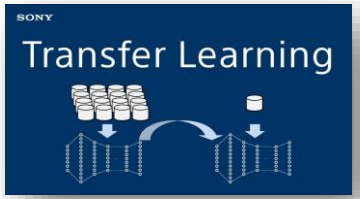
# Designing…




- Kids can draw the letters given section with the given alphabet.
- When clicking the done button then it checks whether the drawn alphabet letter and given alphabet letter are correct or wrong.

# Key Pillars & Relevant Technologies






- Use Convolution Neural Network (CNN) in Deep Learning to recognize and prediction the drawing letters.
- Also, Used the Transfer Learning Method (VGG16)
- Use the image_compare library to check mistakes for the hand-drawn letters.
- Pixel Comparison Algorithm - (IMED – IMage Euclidean Distance)

# Key Pillars & Relevant Technologies

## Hardware Requisites

- Our basic plan is to implement the application operating on an Android device with a drawing platform
  - Mobile Phone

## Technologies to be used

- Flutter
- Python
- TensorFlow

# User Requirement & Functional Requirement

## Functional Requirements

- Recognize hand-drawn letters.
- Predict the letter if the student draws the letter incompletely.
- Identify mistakes done by the student when drawing the letter and correct the mistake letter.

## User Requirements

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

# Challengers & Overcomes

## ❑ Challengers

- Lack of existing literature and works related to the outcome of study.
- Difficult to find most suitable data set for model.
- Difficult to use some parameters when creating a model for the study (optimizes, activations).
- Difficult to intergrade models and mobile applications.
- Difficult to obtain the model outcome in mobile applications.

## ❑ Overcomes

- Follow the guidance of supervisors and experts.
- Refer to the references and previously published works.
- Utilize the theoretical knowledge of the machine learning module.
- Self-studied and used analytical thinking to resolve the problems.

# Overall...

- Completed
  - Developed the 3 models.
  - Completed the frontend of mobile application.
  - Prediction of the alphabet.
  - Connected with the model & android using the tflite library in TensorFlow.
  - Identify the mistakes of the drawn alphabet letter.
  - Recorrect the mistaken letter in a standard way.

# IT19167206 | Wijesinghe S.A.S.D

**B.Sc. (Hons) Degree in Information Technology Specialized in Software Engineering**

# Introduction

- In current education system normally all the modules and syllabus are made for physical environment.
- And primary level education based on teach under proper supervision.
- A situation like pandemic (Corona) or if mentor would not be able to supervise that children's it will not good for them
- And sometimes current Sri Lankan education system teacher is not able to manage all class room students at once

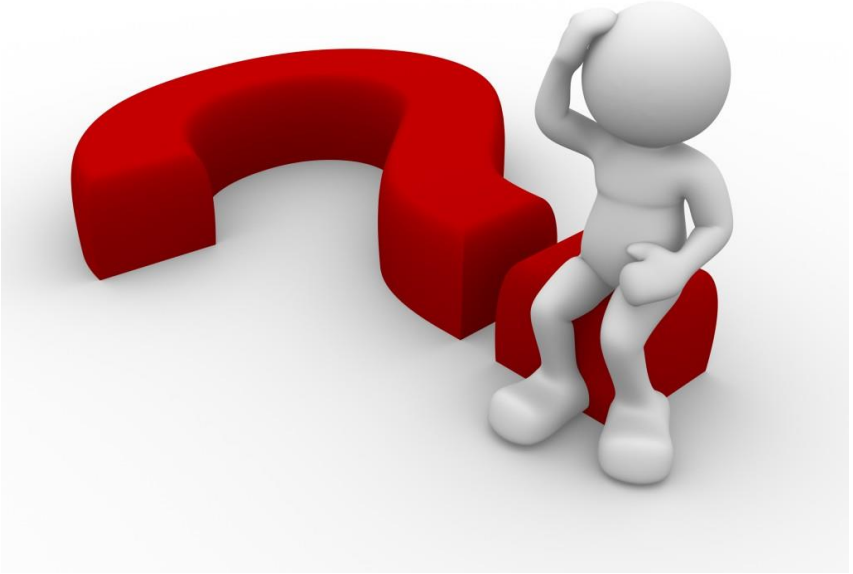
    - Those facts are bring this idea

# Background/Research Gap

- **Image content identification**.
- That’s why those activities are familiarizing in  basic **childhood syllabus**.
- Until today that supervision is **fully manual process**
- If child will have to use method like **distance learning** it will be not effective on their knowledge

# Research Problem

- In primary level student has **developing brain** and most of the time the primary education **base on give to proper understanding** to children about some **basic in our knowledge system,**
- Most of our learning base on some **image understanding** and **context writing**
- In childhood that stage must follow by under any proper **supervision**, If not children will be learning some wrong staff.
- There is **no any automated** way to track or monitor mentioned problem
- So that is the reason world needs some **automated monitoring system** to teach primary students.



# Specific and Sub-Objectives

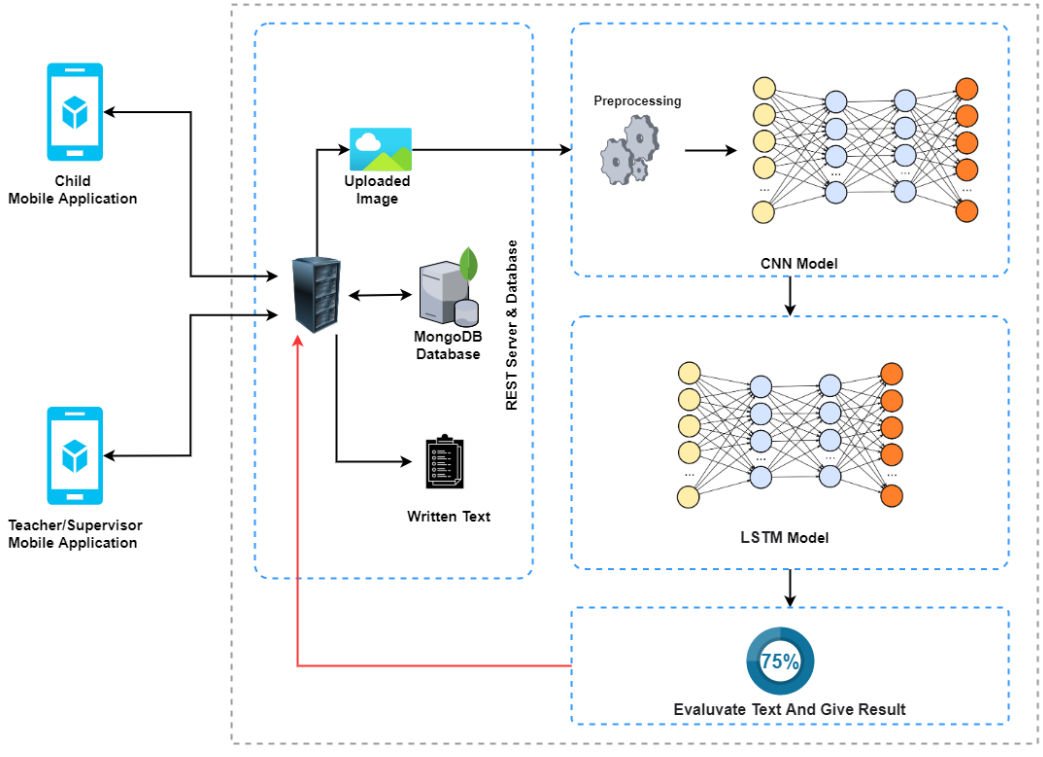
- **Identify uploaded image and generate caption on image compare with student given answers**

  - Identify the image details and generate caption
  - Identify the student answer on writing board.
  - Evaluate those two text and give the result

# Research Methodology

IT19167206 | S.A.S.D. Wijesinghe | 2022-254

# System Diagram



# Methodology

- Image Feature Extraction
  - Use VGG16 model

```
#Extract Image Features
model = VGG16()
model = Model(inputs=model.inputs, outputs=model.layers[-2].output)
```

# Methodology

Image Feature Extraction

Extract Model Image Data

```
features = {}
directory = os.path.join(BASE_DIR, 'Images')

for img_name in tqdm(os.listdir(directory)):
    img_path = directory + '/' + img_name
    image = load_img(img_path, target_size=(224, 224))
    # convert image pixels to numpy array
    image = img_to_array(image)
    # reshape data for model
    image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))
    # preprocess image for vgg
    image = preprocess_input(image)

    # extract features
    feature = model.predict(image, verbose=0)
    image_id = img_name.split('.')[0]
    features[image_id] = feature
```

# Methodology

- Create Word Mapping

```
# create mapping of image to captions
mapping = {}
# process lines
for line in tqdm(captions_doc.split('\n')):
    # split the line by comma(,)
    tokens = line.split(',')
    if len(line) < 2:
        continue
    image_id, caption = tokens[0], tokens[1:]
    # remove extension from image ID
    image_id = image_id.split('.')[0]
    # convert caption list to string
    caption = " ".join(caption)
    # create list if needed
    if image_id not in mapping:
        mapping[image_id] = []
    # store the caption
    mapping[image_id].append(caption)
```

# Methodology

## Data Generator

```
def data_generator(data_keys, mapping, features, tokenizer, max_length, vocab_size, batch_size):
    # loop over images
    X1, X2, y = list(), list(), list()
    n = 0
    while 1:
        for key in data_keys:
            n += 1
            captions = mapping[key]
            # process each caption
            for caption in captions:
                # encode the sequence
                seq = tokenizer.texts_to_sequences([caption])[0]
                # split the sequence into X, y pairs
                for i in range(1, len(seq)):
                    # split into input and output pairs
                    in_seq, out_seq = seq[:i], seq[i]
                    # pad input sequence
                    in_seq = pad_sequences([in_seq], maxlen=max_length)[0]
                    # encode output sequence
                    out_seq = to_categorical([out_seq], num_classes=vocab_size)[0]

                    # store the sequences
                    X1.append(features[key][0])
                    X2.append(in_seq)
                    y.append(out_seq)
            if n == batch_size:
                X1, X2, y = np.array(X1), np.array(X2), np.array(y)
                yield [X1, X2], y
                X1, X2, y = list(), list(), list()
                n = 0
```

# Methodology

## Model Creation

```
# encoder model
# image feature layers
inputs1 = Input(shape=(4096,))
fe1 = Dropout(0.4)(inputs1)
fe2 = Dense(256, activation='relu')(fe1)
# sequence feature layers
inputs2 = Input(shape=(max_length,))
se1 = Embedding(vocab_size, 256, mask_zero=True)(inputs2)
se2 = Dropout(0.4)(se1)
se3 = LSTM(256)(se2)

# decoder model
decoder1 = add([fe2, se3])
decoder2 = Dense(256, activation='relu')(decoder1)
outputs = Dense(vocab_size, activation='softmax')(decoder2)

model = Model(inputs=[inputs1, inputs2], outputs=outputs)
model.compile(loss='categorical_crossentropy', optimizer='adam')
```

# Methodology

## Caption Genaration

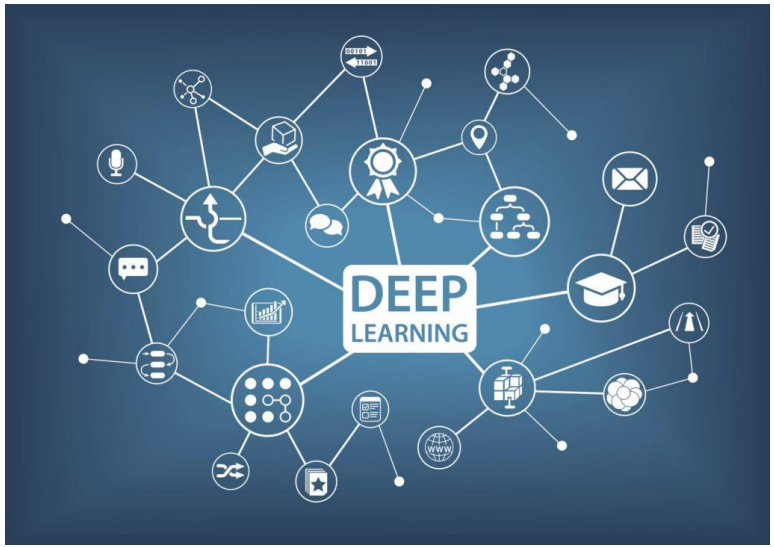
```
# generate caption for an image
def predict_caption(model, image, tokenizer, max_length):
    # add start tag for generation process
    in_text = 'startseq'
    # iterate over the max length of sequence
    for i in range(max_length):
        # encode input sequence
        sequence = tokenizer.texts_to_sequences([in_text])[0]
        # pad the sequence
        sequence = pad_sequences([sequence], max_length)
        # predict next word
        yhat = model.predict([image, sequence], verbose=0)
        # get index with high probability
        yhat = np.argmax(yhat)
        # convert index to word
        word = idx_to_word(yhat, tokenizer)
        # stop if word not found
        if word is None:
            break
        # append word as input for generating next word
        in_text += " " + word
        # stop if we reach end tag
        if word == 'endseq':
            break

    return in_text
```

# Technologies

- Android
- Node JS
- Python
- AWS or Google Storage
- Jupyter Notebook

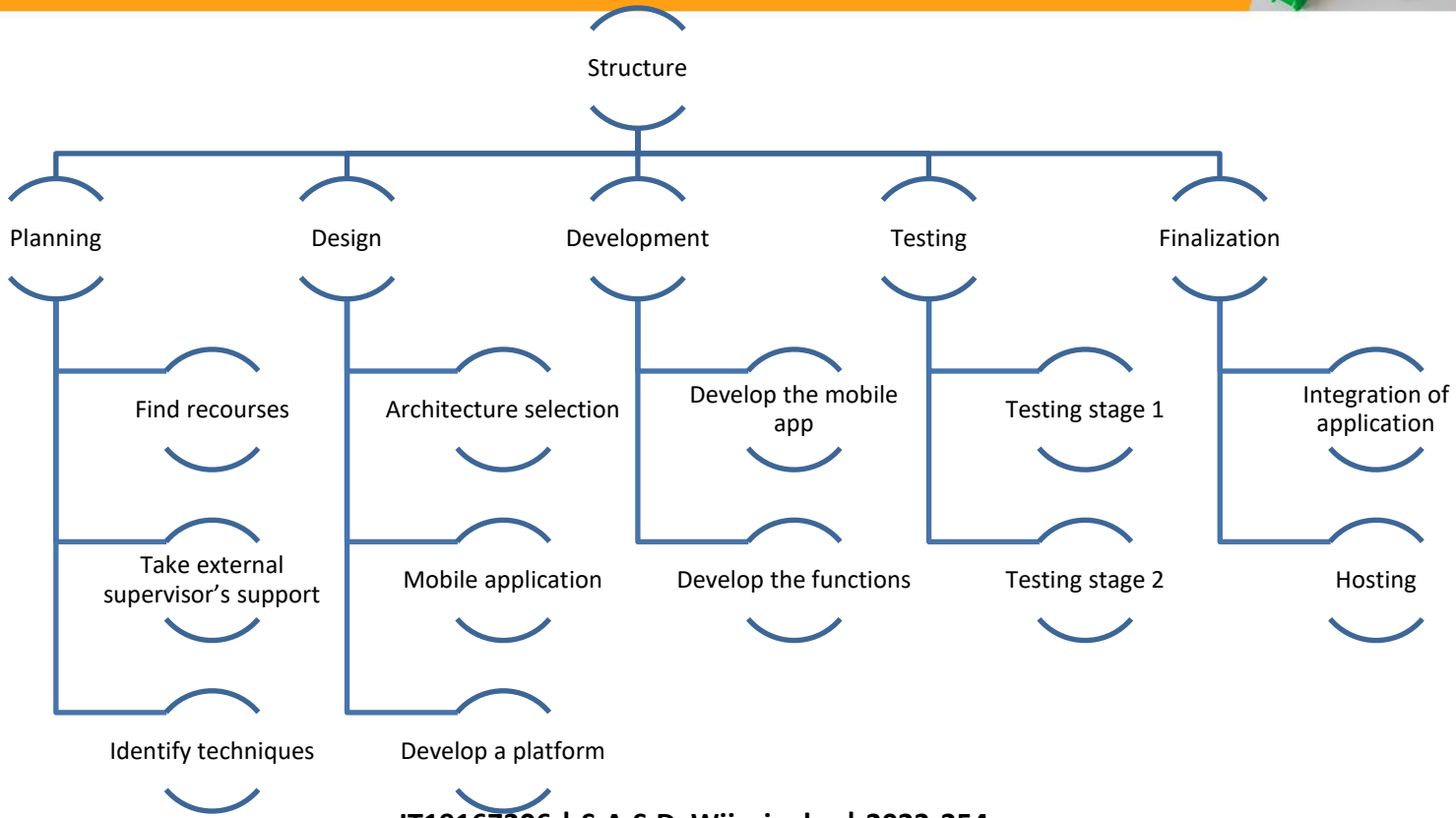
# Key Pillars

# Techniques of Key Pillars




- Deep Learning
  - Neural Network
    - CNN
    - LTSM

# Work Breakdown Structure



# Results

- Depending on the previous research this one gained conspicuous success and, in this model, identify the object and characters in 90% accuracy and identify the action and behaviors on the image in 80% accuracy and this model is capable for make caption more than 70% accurate.

# In 90% Progress

- Completed backend APIs
- Frontend UIs
- Combined Application
- UI testing
- API testing

# Next 10%

- Intergrade With main application
- Hosting APIs in Cloud and Making them Global

# IT19148014|J.T Jayasundara

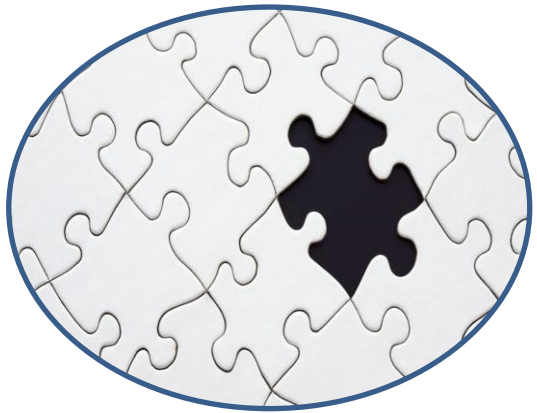
**B.Sc. (Hons) Degree in Information Technology Specialized in Software Engineering**

# Identify hand-drawn shapes and recognize art drawn using shapes

# Research Gap

| Platform | Identify hand-drawn shape | Identify an art drawn using shapes |
|---|---|---|
| Nursery App | ✗ | ✗ |
| Early Learn | ✗ | ✗ |
| Using Fuzzy Logic to Recognize Geometric Shapes Interactively | ✓ | ✗ |
| Sketch Recognition with Natural Correction and Editing | ✓ | ✗ |

# Background/Research Gap



A smart shape-learning method has been invented. The specialty and novelty of this method are students can draw the shape of their t and then the system will identify the drawn shape and suggest corrections if needed. Below functionality also will include in this method.

- Identify an art drawn using shapes by the student.

# Research Problem

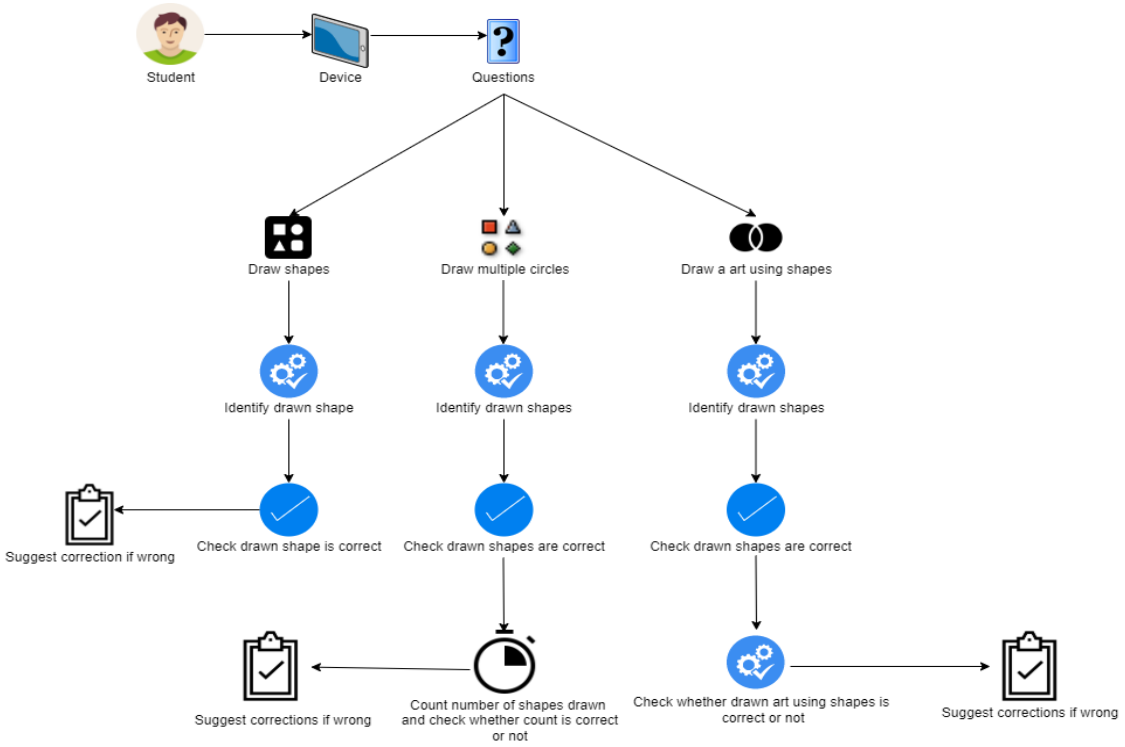
- Students' mind in primary education is very sensitive. If they think that some learning is hard to do and I cannot do this, that learning will never be easy for the students.
- Mental problems can be affected for students because of hard teaching methods.
- There are very short amount of application in the world that can identify hand drawn shapes.
- Primary age children's mindsets are willing to study with the drawing and pictures. But there is no better method to do that.

# Objectives

- Identify hand-drawn shapes by primary ages kids.
- Identify hand-drawn arts that can draw using shapes by primary ages kids.
- Provide correct incorrect status of shapes and arts for kids

# Research Methodology

# System Diagram

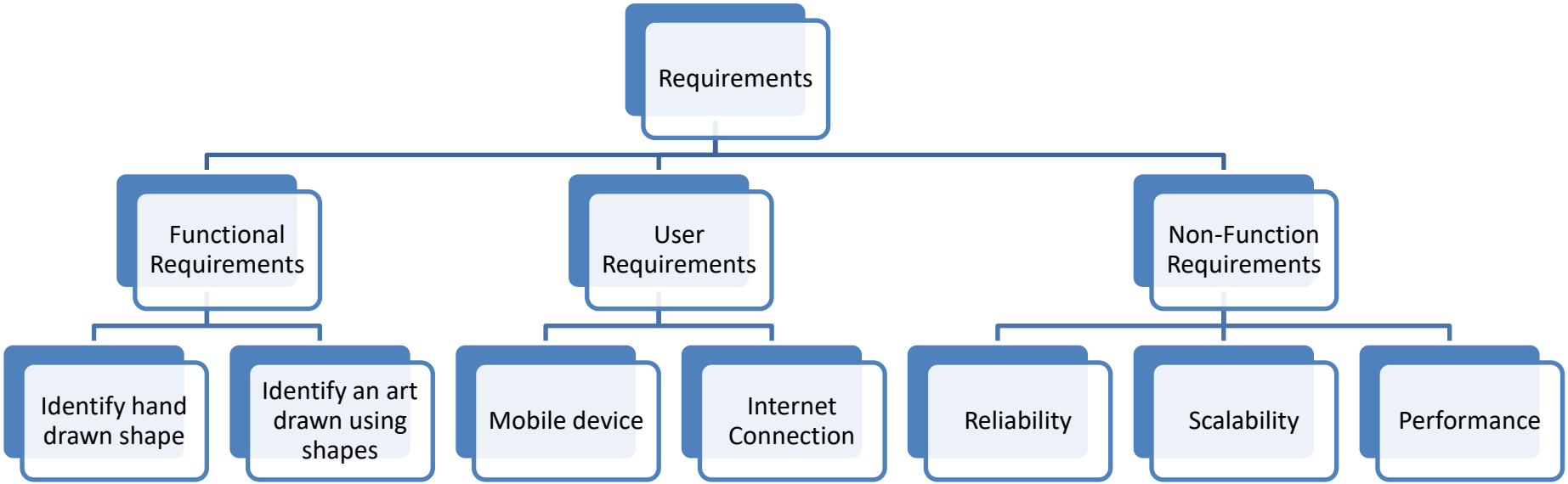


# Technologies

- Python
- Flutter
- Dart

# Key Pillars

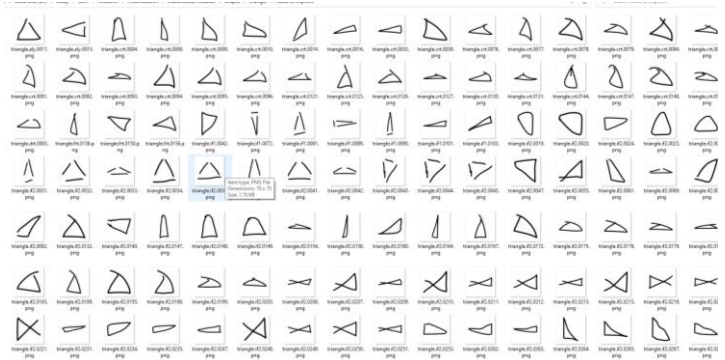
# Requirements



# Implementation..

- Dataset

  - The dataset was created by primary ages kids
  - *Contain 5033 hand drawn images in dataset.*



# Implementation..

- Steps to create the model

  - Data preparation
    - Load data
    - Reshape
    - Split data into train & test
  - Create CNN based model
    - Define the model
    - Set the activation & optimizer
  - Evaluate the model
    - Train for identify correct and incorrect shapes

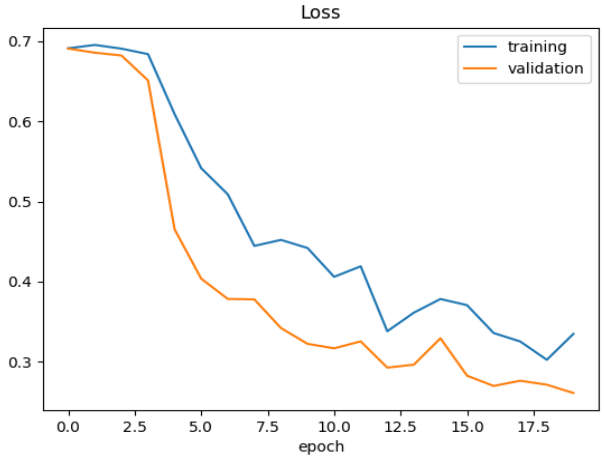
# Implementation..

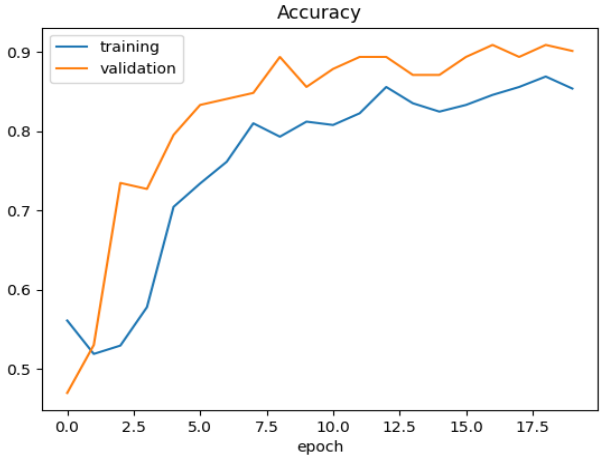
## ❑ Model Creation

```
def cn_model():
    no_of_filters = 60
    size_of_filter_1 = (5, 5)
    size_of_filter_2 = (3, 3)
    size_of_pool = (2, 2)
    no_of_node = 500

    model = Sequential()
    model.add((Conv2D(no_of_filters, size_of_filter_1, input_shape=(image_dimension[0],
                                                                    image_dimension[1], 1), activation='relu')))
    model.add((Conv2D(no_of_filters, size_of_filter_1, activation='relu')))
    model.add(MaxPooling2D(pool_size=size_of_pool))
    model.add((Conv2D(no_of_filters // 2, size_of_filter_2, activation='relu')))
    model.add((Conv2D(no_of_filters // 2, size_of_filter_2, activation='relu')))
    model.add(MaxPooling2D(pool_size=size_of_pool))
    model.add(Dropout(0.5))

    model.add(Flatten())
    model.add(Dense(no_of_node, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(no_of_data_category, activation='softmax'))

    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
    return model
```

- 11 layers have used for this model.
- Used ReLU for activation to get the positive result.
- Used Softmax activation for multi-class classification problems where required on more than two class labels.

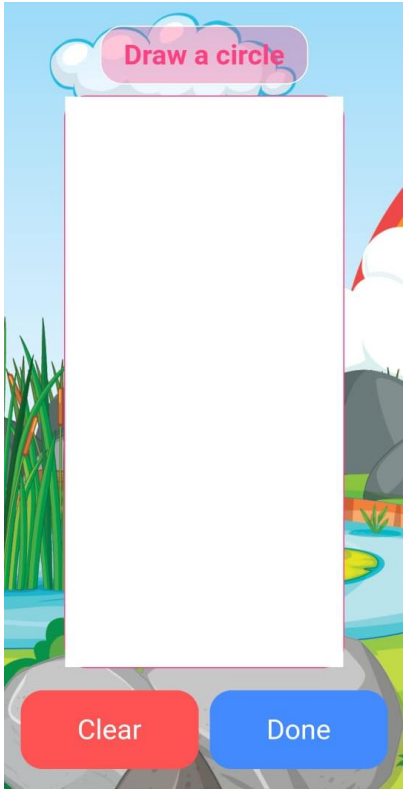
# Result

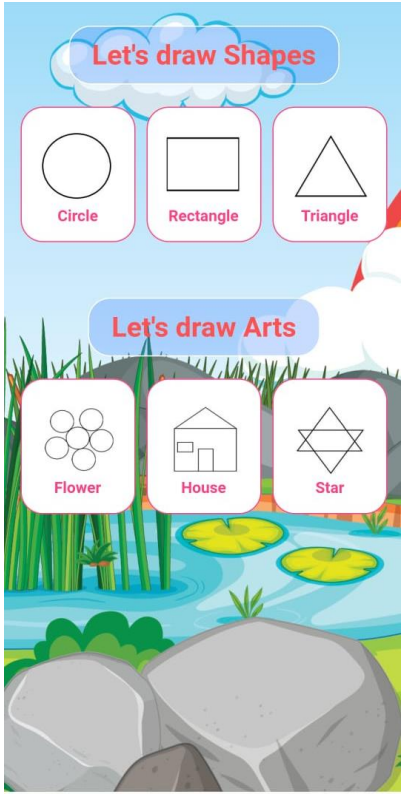


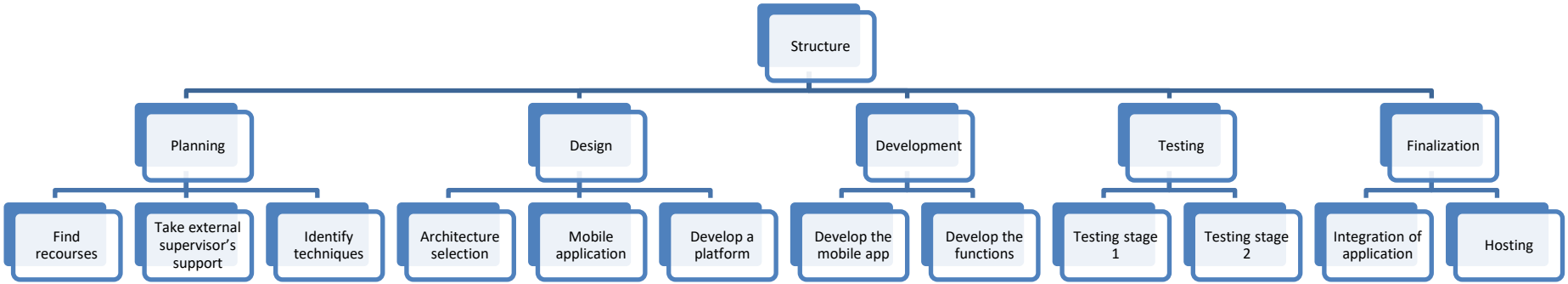

- This model has more than 87% accuracy

# Design

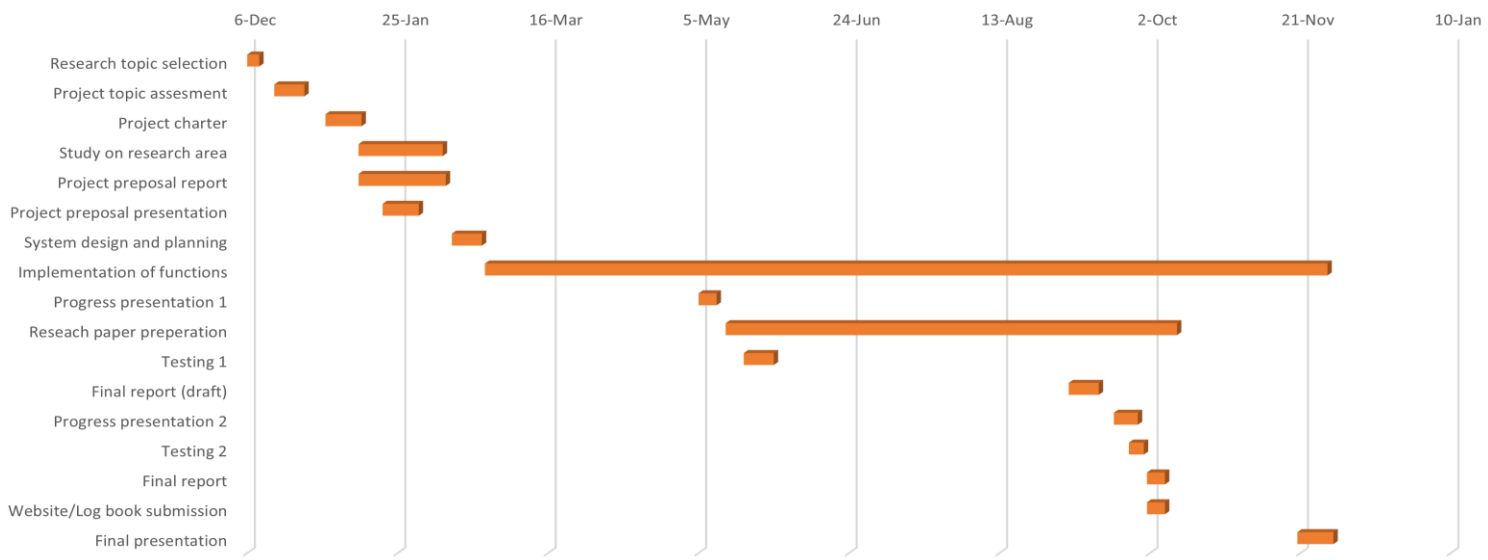

Draw a circle
Clear
Done


Let's draw Shapes
Circle
Rectangle
Triangle
Let's draw Arts
Flower
House
Star

# Work Breakdown Structure



# Grant Chart


6-Dec
25-Jan
16-Mar
5-May
24-Jun
13-Aug
2-Oct
21-Nov
10-Jan
Research topic selection
Project topic assesment
Project charter
Study on research area
Project preposal report
Project preposal presentation
System design and planning
Implementation of functions
Progress presentation 1
Reseach paper preperation
Testing 1
Final report (draft)
Progress presentation 2
Testing 2
Final report
Website/Log book submission
Final presentation

# IT19197838|L.H.G.N.Ravindu

**B.Sc. (Hons) Degree in Information Technology Specialized in Information Technology**

# Introduction

- The preschool years are a critical period for children to receive high-quality personal care and learning experiences.
- Many primary schools still use the **traditional learning process** since it is very common.
- In a typical classroom, basic materials such as a **blackboard, whiteboard, chalk, and marker** are used.
- Our program allows students to recognize **objects and colors**.

# Research Problem

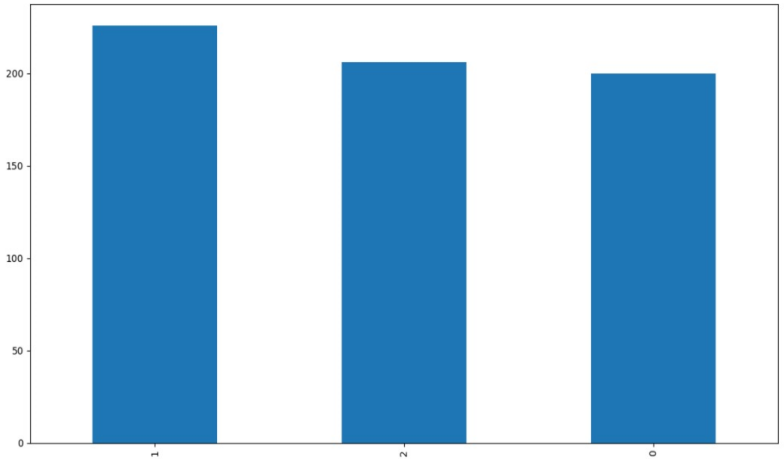
- Traditional education places a greater emphasis on teaching than on learning. There has been a lot of work put into developing a basic improvement that will ensure successful learning. Students used traditional methods such as color pencils, watercolors, papers, and other learning materials in this scenario. This is a massive waste. Also, most students are uninterested in this material, but they really interact with mobile phones, so we provide mobile solution for interactive learning for primary students.

# Objectives

- Determine what pictures students drew on the surface.
- Identify mistakes done by student
- The algorithm will recognize the color of the student's sketched picture.
- Make a color suggestion that is appropriate.
- Testing and checking the accuracy.
- Finally, recommend a student's drawn object corrections.

# Implementation

- Dataset




- More than 600 images drawn by pre-school students.
- All these pictures were drawn by children of two preschools.

# Implementation

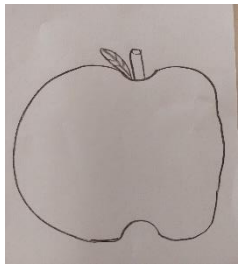
- Data collection

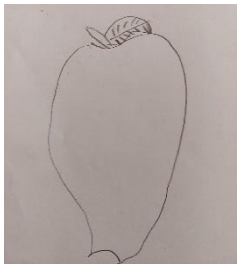
Little Kids Pre-School

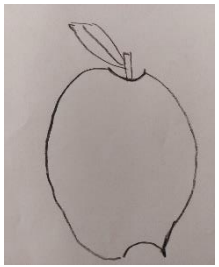
Brilliants Way Pre-School

# Implementation

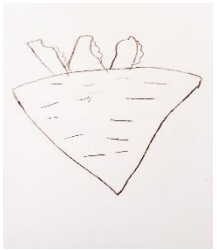
Apple



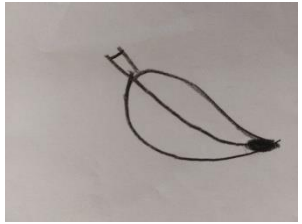


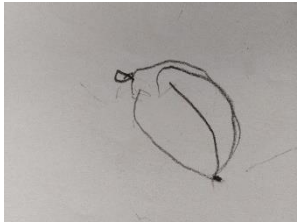


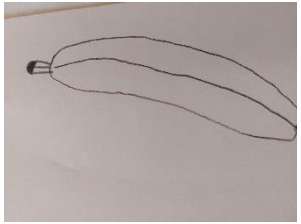
Carrot



Banana







# Implementation

- Model Creation

```
model = Sequential()

model.add(Conv2D(256, (3, 3), input_shape=data.shape[1:]))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(128, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())

model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))

model.add(Dense(4, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

model.load_weights('model.h5')
```

•Use MaxPooling 2d , Conv2D, Dropout, Relu Function and Dense Layers

# Technologies

Keras

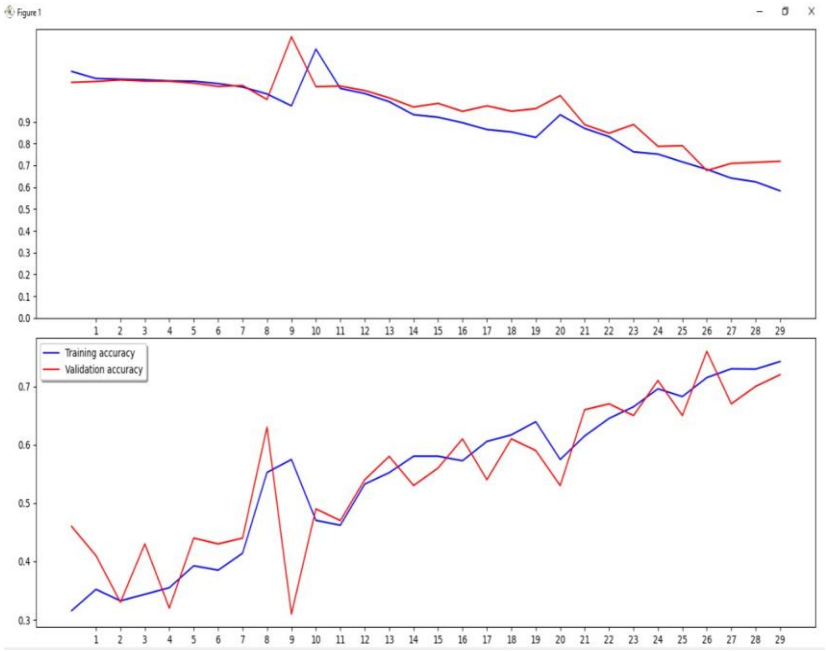
Tenserflow

Flask

OpenCV

Flutter

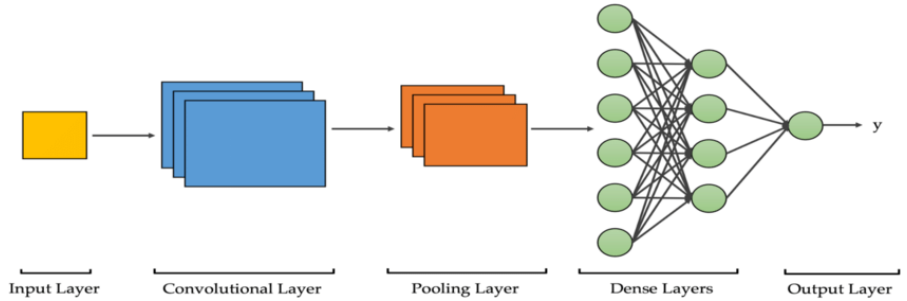
# Result



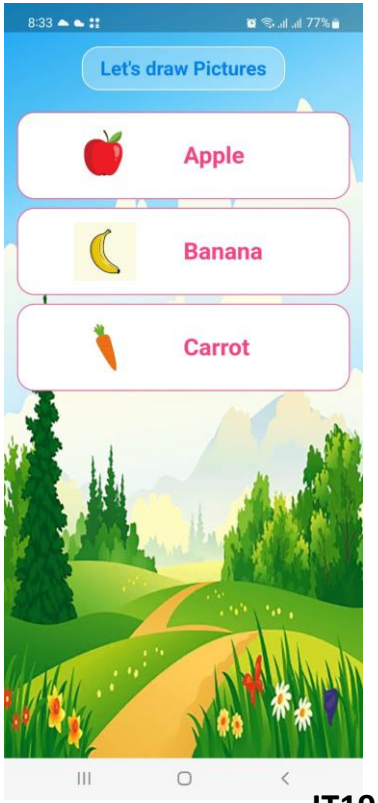

- More than 89% accuracy.

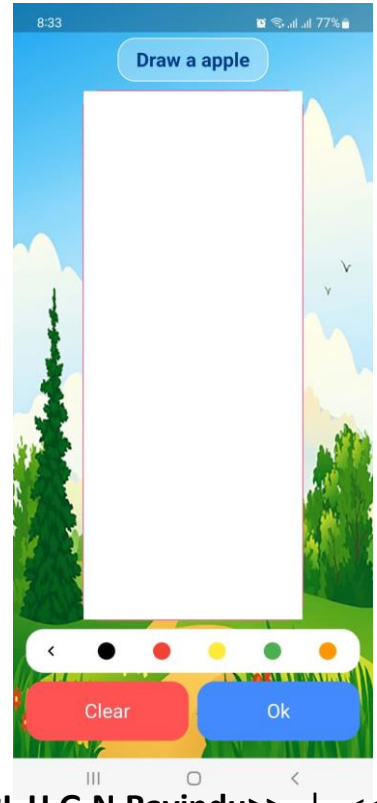
# Key Pillars & Relevant Technologies

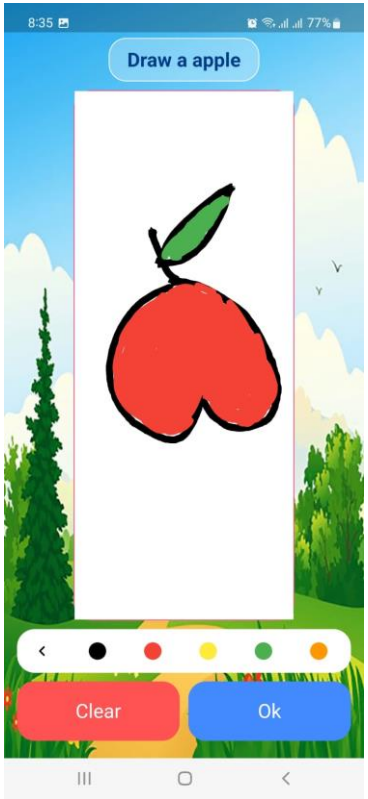
- Use Convolution Neural Network (CNN) to detect drawing objects.
- A subclass of deep neural networks is CNN.
- In terms of recognizing and segmenting certain objects, CNN has shown better results.



# Design


Let's draw Pictures
Apple
Banana
Carrot


Draw a apple
Clear
Ok

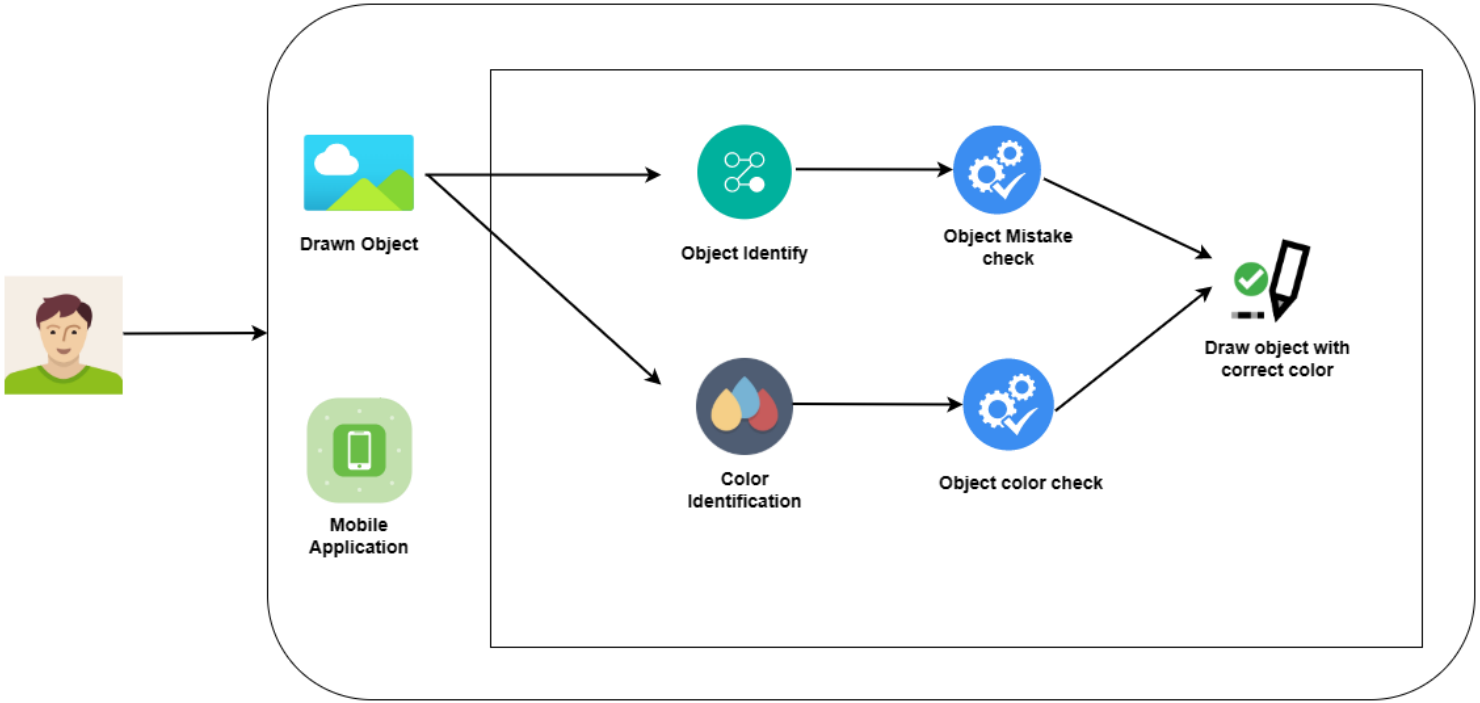

Draw a apple
Clear
Ok

# Demo

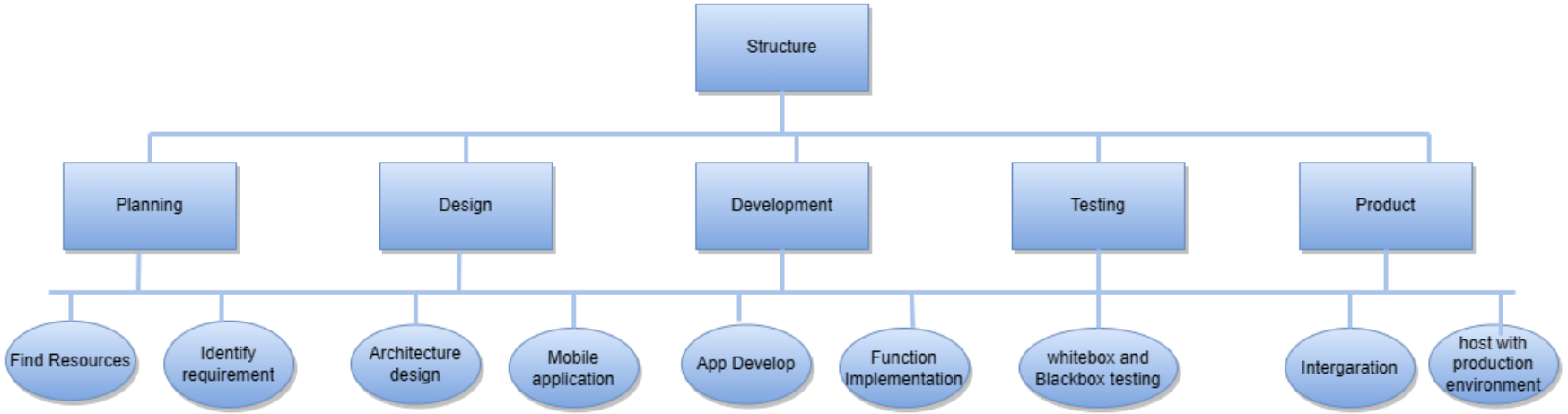
# In Final Presentation

- Improve with the primary application
- API hosting in the cloud and globalization

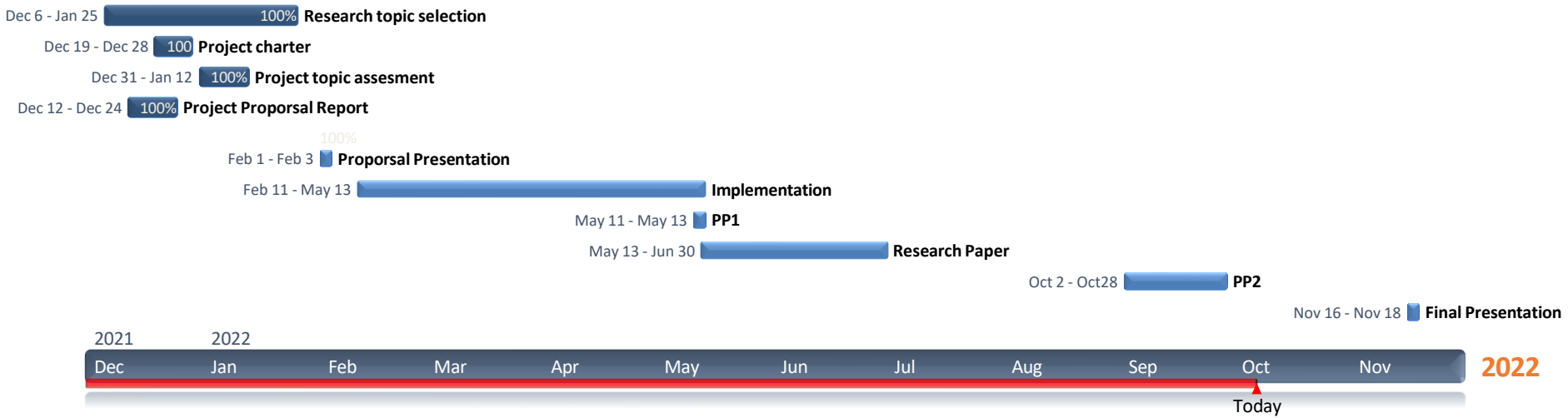
# System Architecture



# Work Breakdown Structure



# Grant Chart


Dec 6 - Jan 25
100%
Research topic selection
Dec 19 - Dec 28
100
Project charter
Dec 31 - Jan 12
100%
Project topic assesment
Dec 12 - Dec 24
100%
Project Proporsal Report
100%
Feb 1 - Feb 3
Proporsal Presentation
Feb 11 - May 13
Implementation
May 11 - May 13
PP1
May 13 - Jun 30
Research Paper
Oct 2 - Oct28
PP2
Nov 16 - Nov 18
Final Presentation
2021
2022
Dec
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
2022
Today

# COMMERCIALIZATION

# Our App

- **The First Computerized Commercialized & Automated Primary Education Application in Sri Lanka**
- **First App Introduce For Primary School Students Distance Learning**
- **75% Automated Flow For Teachers**

## Main Features

- **Teach students letters efficiently.**
- **Improve the ability to draw images.**
- **Increasing Knowledge of shapes.**
- **Improve the ability to identify features in images.**

# MARKET

How many children are in school in Sri Lanka?

According to the Ministry of Statistics, today there are approximately **10,012 public** schools serving close to 4,037,157 students, all around the island.

- Mean there will be around 1.5 Million primary students
- Approx. Customer Base 0.5 Million

# Flyer


Teach Your Kids
With Our App
Interactive Learning System For Kids
Coming Soon....
Interactive
Learning
For Kids
Draw Alphabet
Draw Shape
Identify Object
Write Sentences

## How We Can Promote This App ?

- Facebook Advertisement
- Referral Program
- Notice
- TV Commercial
- Flyers

# How Make Money This App ?

- User Subscriptions
- Partnerships

## Free Plan

What You'll Get

- Unlimited Access
- Ads Free
- Notify the child errors
- Gold Member Card

**Free**

Try Premium For 1 Month Free

Selected

## Premium Plan

What You'll Get

- Unlimited Access
- Ads Free
- Notify the child errors
- Gold Member Card

**RS 299.00**/month

Choose

## Enterprise Plan

What You'll Get

- Unlimited Access
- Ads Free
- Notify the child errors
- Gold Member Card

**Negotiable**

Choose