

# 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

# 01 Introduction

## 02 Problem Definition

## 03 Objectives

## 04 Individual Contribution

## 05 Commercialization



# INTRODUCTION

- The education field has been revolutionized by introduction of information technology-based tools as a teaching aid.
- The effective integration of educational technology and early childhood education requirements are limited.
- The primary aim of this study 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



# Problem Definition

- Today's generation is more interested in learning from an interactive computer or mobile application-based activities.
- But limited number of education systems have developed as automatic self learning platforms.
- The COVID-19 pandemic created space between children and manual supervision by teachers.
- Parents might not have sufficient time or knowledge to assist to enhance primary level educational concepts.
- This interactive learning system providing an innovative real-time learning experience for children that does not require any manual supervision.



# 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



- Mainly aims to identify and predict hand-drawn letters, in order to provide suggestions to recorrect the letter in professional manner.
- Existing systems and applications are allowed children to draw a letter on a provided letter-shaped image.
- But present research work provide by
  - Allowing children to write letters on screen freely
  - predicting the drawing as letter
  - identifying mistakes in the basic letter
  - providing correct way to draw the letter
- This system will be provided self learning platform to children to improve alphabetical knowledge and skills as unique learning experiences.



# Implementation..



## ❑ Dataset

### **A-Z Handwritten Alphabets in .csv format**

370000+ English Alphabets Image Data-set



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

# Implementation..



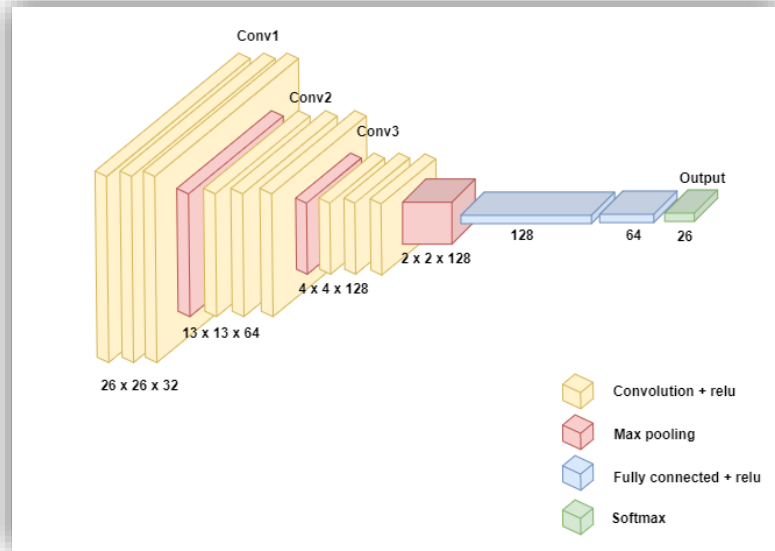
## ❑ Steps to create the model

- ❖ Data preparation
  - Load data
  - Reshape
  - Split data into train & test
  - Scale data
- ❖ Create CNN based model
  - Define the model
  - Set the activation & optimizer
- ❖ Evaluate the model
  - Training & validation curves
- ❖ Prediction

# Implementation..



## ❑ Model Creation



- Use three 2D Convolution layers.
- Every Conv2D layers have max pooling operation to extract maximum value from the Feature map according to filter size.

# Implementation..



## ❑ Model Creation

```
# 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

```
# 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

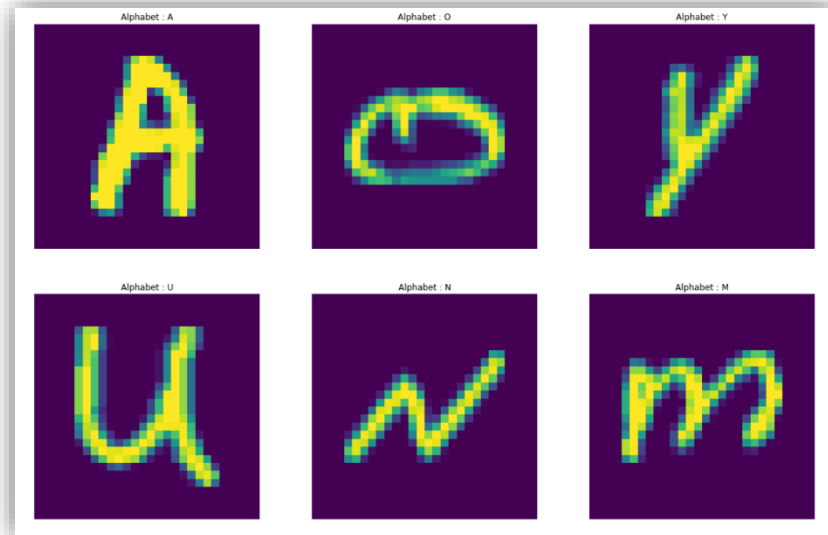
model.add(Dense(128, activation="relu"))
model.add(Dense(64, activation="relu"))
# output layer
model.add(Dense(26, activation="softmax"))
# compile
model.compile(optimizer=adam_v2.Adam(learning_rate=0.001),
              loss='categorical_crossentropy', metrics=['accuracy'])
# model summary
model.summary()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

- 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..



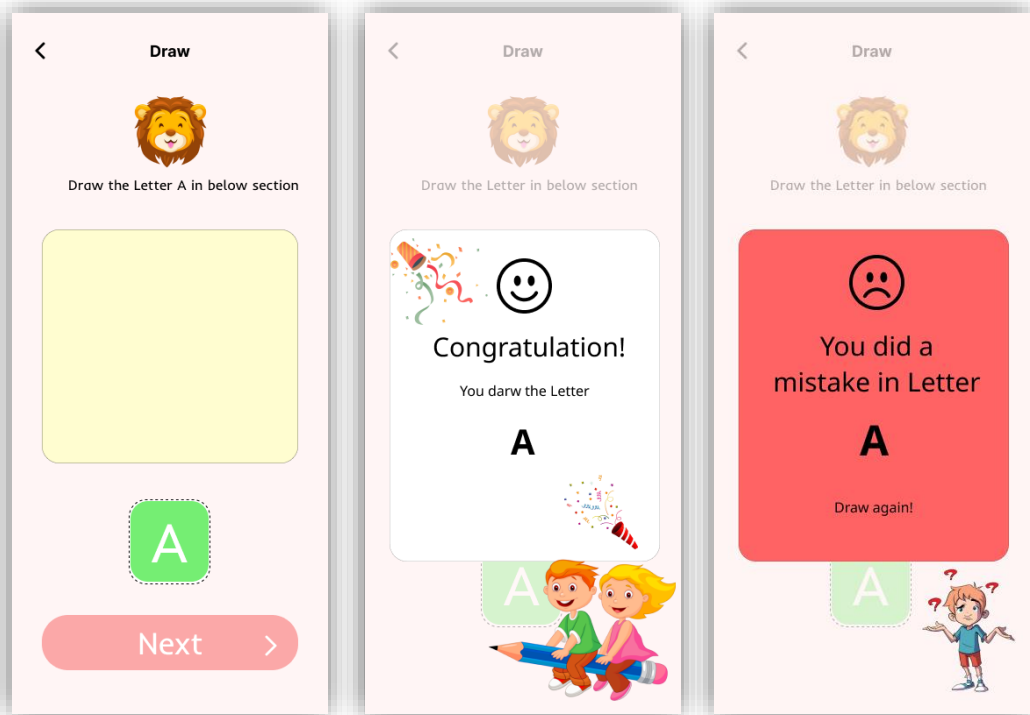
## ❑ Training data



*Training data*



# Designing



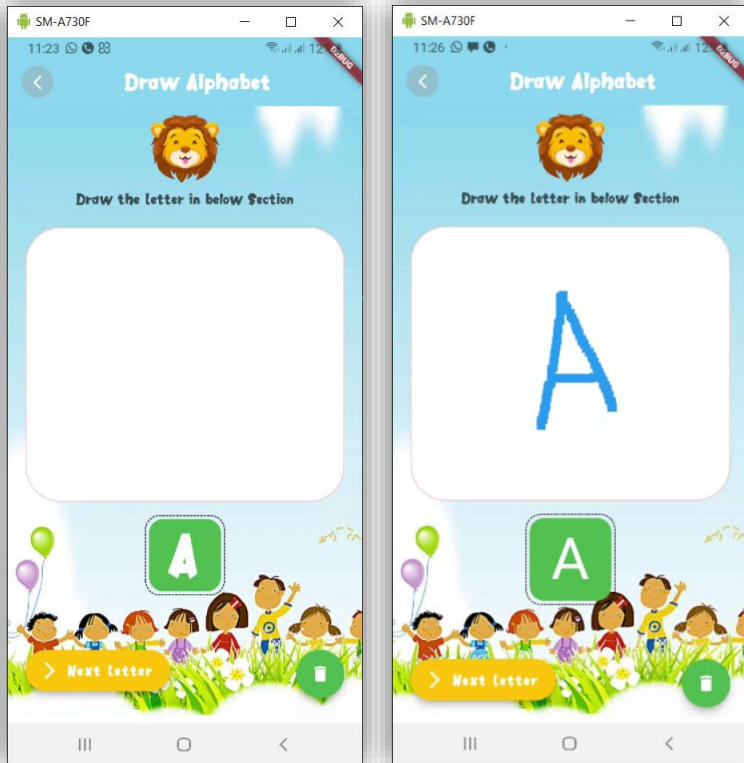
- Use Figma tool to design the Mobile Interface.
- Here shows pre-design interfaces.

# 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 clicked the next alphabet button then moves to the next alphabet letter.

# Key Pillars & Relevant Technologies



- Use Convolution Neural Network (CNN) in Deep Learning to recognize and prediction the drawing letters.
- CNN is a class of Deep Neural Network.
- CNN has achieved better result in detecting and segmenting the specific objects.

# 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

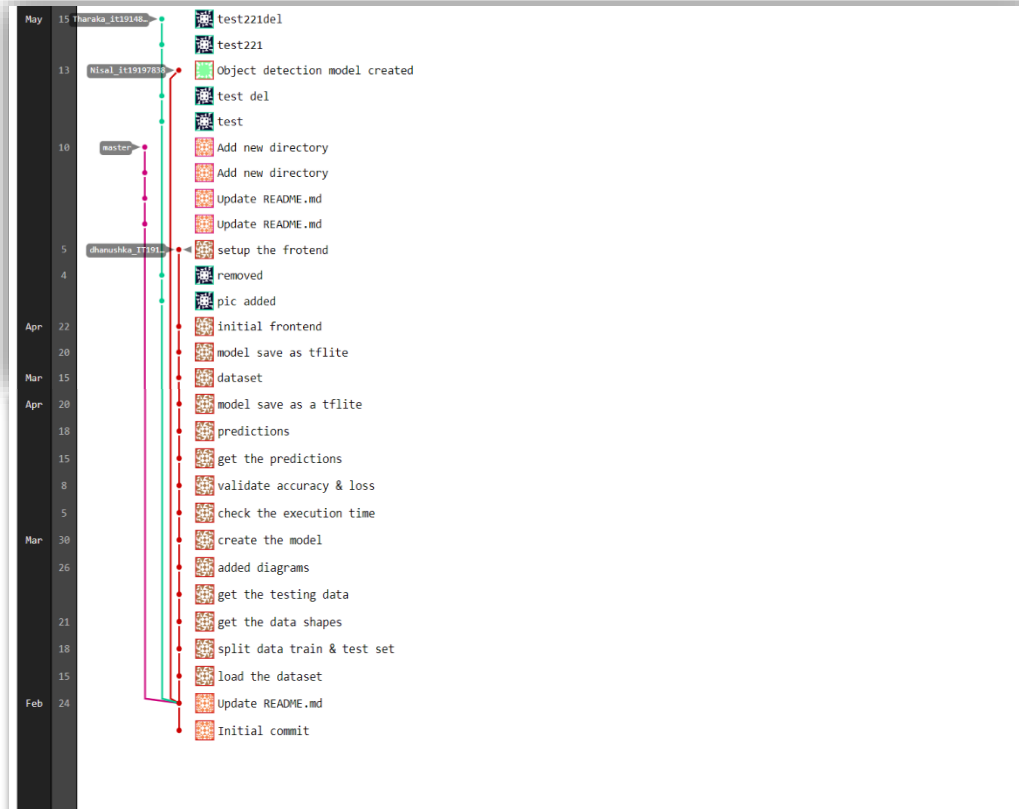
- Recognize hand-drawn letters.
- Predict the letter if the student draws the letter incompletely.
- Identify mistakes done by the student when drawing the letter and correct the mistake letter.

## User Requirements

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



# Version Control Mechanism



# 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...



## ☐ Done

- Completed the developing model.
- Completed the Frontend of mobile application.
- Prediction of the alphabet.
- Connected with the model & android using the tflite library in TensorFlow.

## ☐ To be done

- Identify the mistakes of the drawn alphabet letter.
- Correctness alphabet letter of mistake letter.
- Try to adjust the model with changing the optimizers, activations, loss functions & other specific functions to check the better output result.



# IT19148014 | J.T Jayasundara

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

# OBJECTIVES



- Identify and classify hand-drawn shapes
- Give grades to hand-drawn shapes
- Identify and classify hand-drawn arts that are drawn using shapes
- Give grades to hand-drawn arts
- Count the number of shapes in the art

# What I have done for fifty percent of the project



- Created a dataset
- Create a model to classify hand-drawn shapes
- Identified and classified hand-drawn shapes
- Gave grades to drawn shapes



# Research Problem



- Lack of accurate hand-drawn shape identification application systems.
- Even though drawing shapes is very important in primary education, there is no application to give grades for hand-drawn shapes.



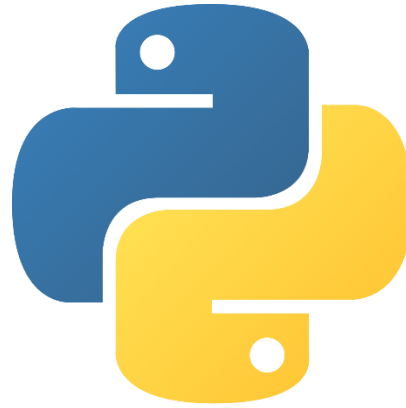
# Research Methodology



# Technologies



- Python
- Flutter



# Key Pillars



Image  
Processing

Deep  
Learning



# Dataset



Name

Date modified

Type

Size

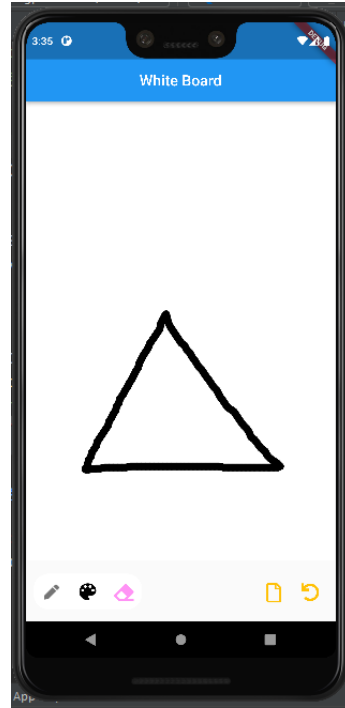
0  
1  
2  
3  
4  
5

5/4/2022 12:51 PM  
5/4/2022 12:51 PM  
5/4/2022 12:51 PM  
5/4/2022 12:51 PM  
5/4/2022 12:51 PM  
5/4/2022 12:51 PM

File folder  
File folder  
File folder  
File folder  
File folder  
File folder



# Frontend





# Results



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help PSE [D:\study\SLIIT\Research\PrimaryEducation\Project\RealProject\PSE\PSE] - Test.py

PSE > Part_01 > Test.py

Project
└─ PSE
   └─ Part_01
      └─ Data
         ├── 0
         ├── 1
         ├── 2
         ├── 3
         ├── 4
         └── 5
      └─ Test

Classifier.py Test.py

10
11
12
13
14
15
16
17
18 def test(name):
19     model = load_model('model.h5')
20     img = cv2.imread('Test/' + name)
21     # img = cv2.imread('Test/' + name)
22     img = np.asarray(img)
23     img = cv2.resize(img, (32, 32))
24     img = preProcessing(img)
25
26     test()

Run: Test
2022-05-24 22:40:59.197133: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic library 'cusolver64_10.dll'; dlerror: cusolv
2022-05-24 22:40:59.198193: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic library 'cuspars64_11.dll'; dlerror: cuspar
2022-05-24 22:40:59.199254: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic library 'cudnn64_8.dll'; dlerror: cudnn64_8.
2022-05-24 22:40:59.199405: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1757] Cannot dlopen some GPU libraries. Please make sure the missing libraries m
Skipping registering GPU devices...
2022-05-24 22:40:59.203025: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (o
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2022-05-24 22:40:59.204840: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1261] Device interconnect StreamExecutor with strength 1 edge matrix:
2022-05-24 22:40:59.204963: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1267]
2022-05-24 22:40:59.205039: I tensorflow/compiler/jit/xla_gpu_device.cc:99] Not creating XLA devices, tf_xla_enable_xla_devices not set
C:\Users\janit\anaconda3\envs\shape_identification\lib\site-packages\tensorflow\python\keras\engine\sequential.py:450: UserWarning: 'model.predict_classes()' is
warnings.warn('model.predict_classes()' is deprecated and '
2022-05-24 22:40:59.605531: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR optimization passes are enabled (registered 2)
2

Process finished with exit code 0
|

Run TODO Problems Terminal Python Console
PyCharm 2022.1.1 available // Update... (a minute ago) 30:1 CRLF UTF-8 4 spaces Python 3.8 (shape_identification) Event Log
```

# Future Plans



- Create a model to identify and classify arts that are drawn using shapes
- Count the number of shapes includes in art
- Give grades to arts that are drawn using shapes



# Challengers and Overcome



## Challengers

- Lack of existing application to identify and grade hand-drawn shapes
- difficult to find a dataset with grade based

## Overcomes

- Created my own architecture and created an application
- Created my own dataset



# 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 be good for them
  - And sometimes current Sri Lankan education system teacher is not able to manage all classroom students at once
- Those facts are bringing this idea



# Objectives



- Identify the given image
- Generate image caption
- Take Student given answers
- Check given and generated sentence similarity
- Give feedback using similarity

# What I have done for fifty percent of the project



- Image Caption Generator
- Sentence similarity Checker



# Implementation



ADITYAJN105 · UPDATED 2 YEARS AGO

▲ 148

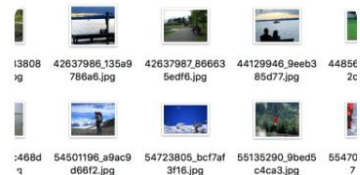
New Notebook

Download (1 GB)



## Flickr 8k Dataset

Flickr8k Dataset for image captioning.



Data

Code (94)

Discussion (0)

Metadata

## About Dataset

### Context

A new benchmark collection for sentence-based image description and search, consisting of 8,000 images that are each paired with five different captions which provide clear descriptions of the salient entities and events. ... The images were chosen from six different Flickr groups, and tend not to contain any well-known people or locations, but were manually selected to depict a variety of scenes and situations

### Usability ⓘ

7.50

### License

CC0: Public Domain

### Expected update frequency

Not specified



# Implementation



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

•
• 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
```

# Implementation



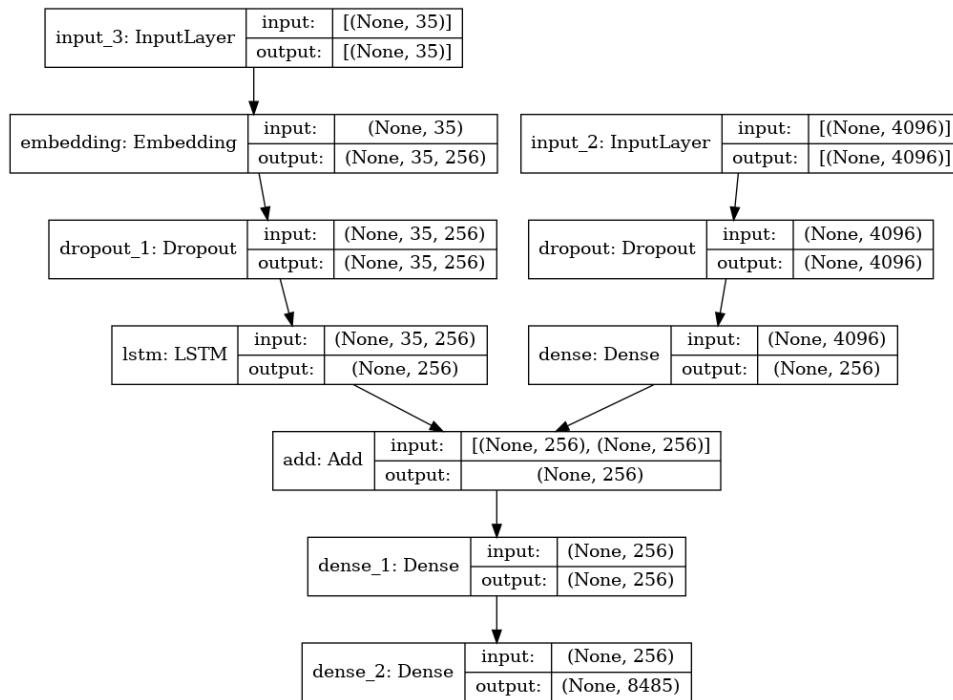
## Model Creation

```
[16]: # 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)

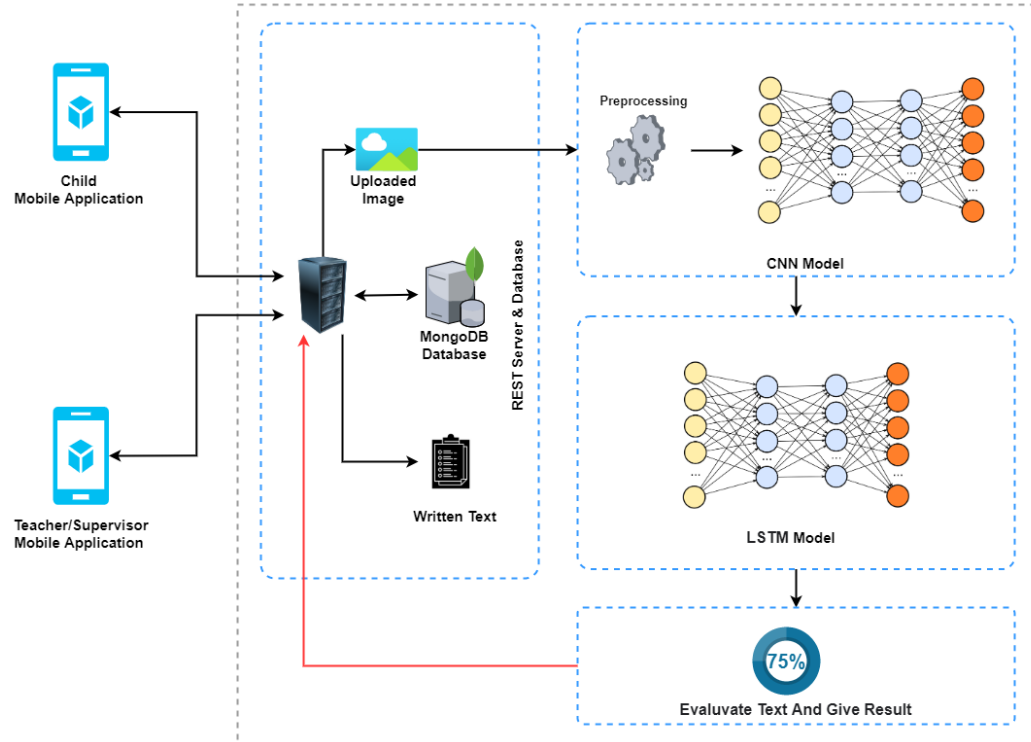
      # decoder model
      decoder1 = add([fe2, se3])
      decoder2 = Dense(256, activation='relu')(decoder1)
      outputs = Dense(vocab_size, activation='softmax')(decoder2)

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

# Implementation



# System Diagram



# Technologies & Key Pillars



## Technologies

- Flutter
- Node JS
- Python
- AWS or Google Storage

android 



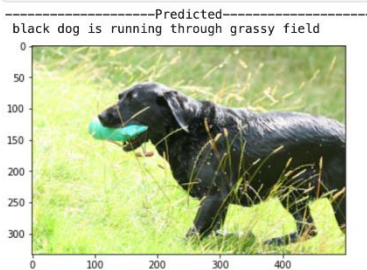
## Key Pillars

- Neural Network
  - CNN
    - conventional neural network
  - LSTM
    - Long Short-Term Memory

# Results

[29]:

```
generate_caption("1026685415_0431cbf574.jpg")
```



▷

```
generate_caption("1030985833_b0902ea560.jpg")
```



0s

```
1 similarity = cosine_similarity(  
2     [sentence_embeddings[0]],  
3     sentence_embeddings[1:]  
4 )  
5 matches = []  
6 for i,sentence in enumerate(sentences[1:]):  
7     print(f"{sentence} \t\t {similarity[0][i]}")  
8     if similarity[0][i] > 0.79:  
9         matches.append(sentence)  
10  
11 print(matches)
```

```
the boy plays football          0.38733065128326416  
the man plays cricket in the playground          1.000000238418579  
what is null safety in dart          0.12113730609416962  
A girl plays cricket          0.6203440427780151  
a dog running on the grass          0.44165387749671936  
black and white dog is running through the grass          0.42550086975097656  
['the man plays cricket in the playground']
```

# Challengers & Overcomes



- Challengers

- Difficulty to find good data set
- Difficulty to create Model for caption generation
- When run the model and other things those cloud recourses are not enough

- Overcomes

- After doing my research I was able to find good data set
- I was able to create separate layer lines for each one and finally I was able to create one
- I hope to but cloud instance to host this all things



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

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

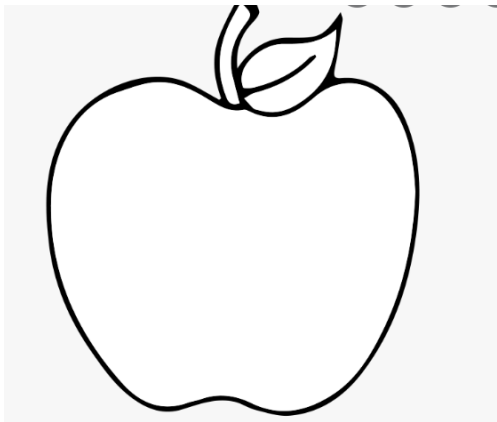


# OBJECTIVES



- Recognize hand-drawn objects.
- Color identification in the relevant object and matching real-life colors.
- Identify mistakes that drawn object and suggest color if wrong
- Draw the right object with the correct color

# The Process so far..



Input

```
<frozen importlib._bootstrap>:219: RuntimeWarning: scipy._lib.  
2022-05-26 00:00:22.895339: I tensorflow/core/platform/cpu_fea  
To enable them in other operations, rebuild TensorFlow with th  
[[9.999995e-01 2.441478e-09 4.958790e-07 9.543984e-10]]  
[0]  
0.9999995  
  
Process finished with exit code 0
```

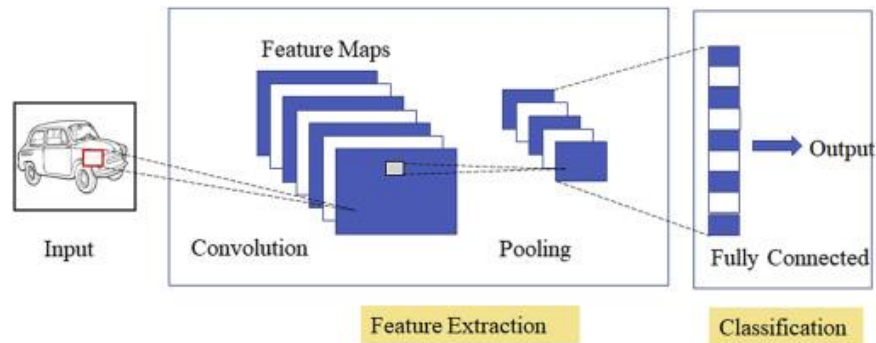
Output

❖ Used CNN for object detection

# CNN



- What is CNN ?
- Why did it use?
- What are the pros and cons?



# Weightage of Component



1 

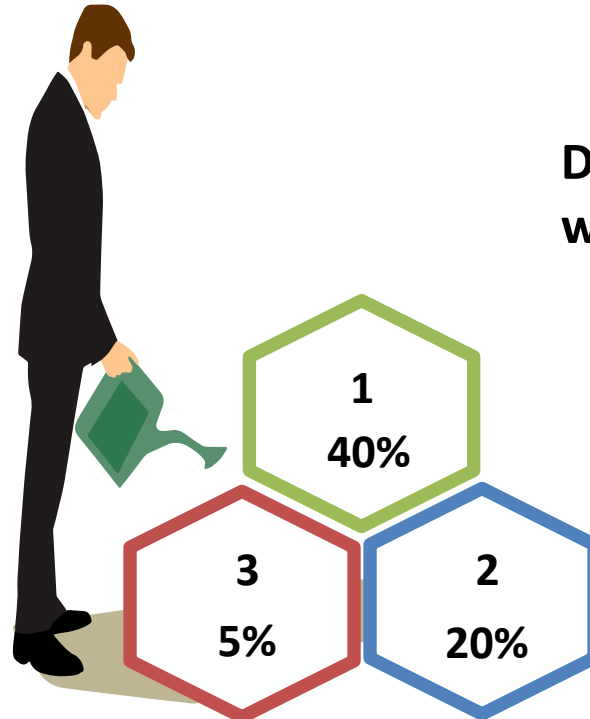
**Hand drawn Object  
detection**

2 

**Color identification  
and match relevant  
color**

3 

**Draw the right object  
with the correct color**



# Completion Rate



## Current Progress

- Model for object detection

- Color identification process

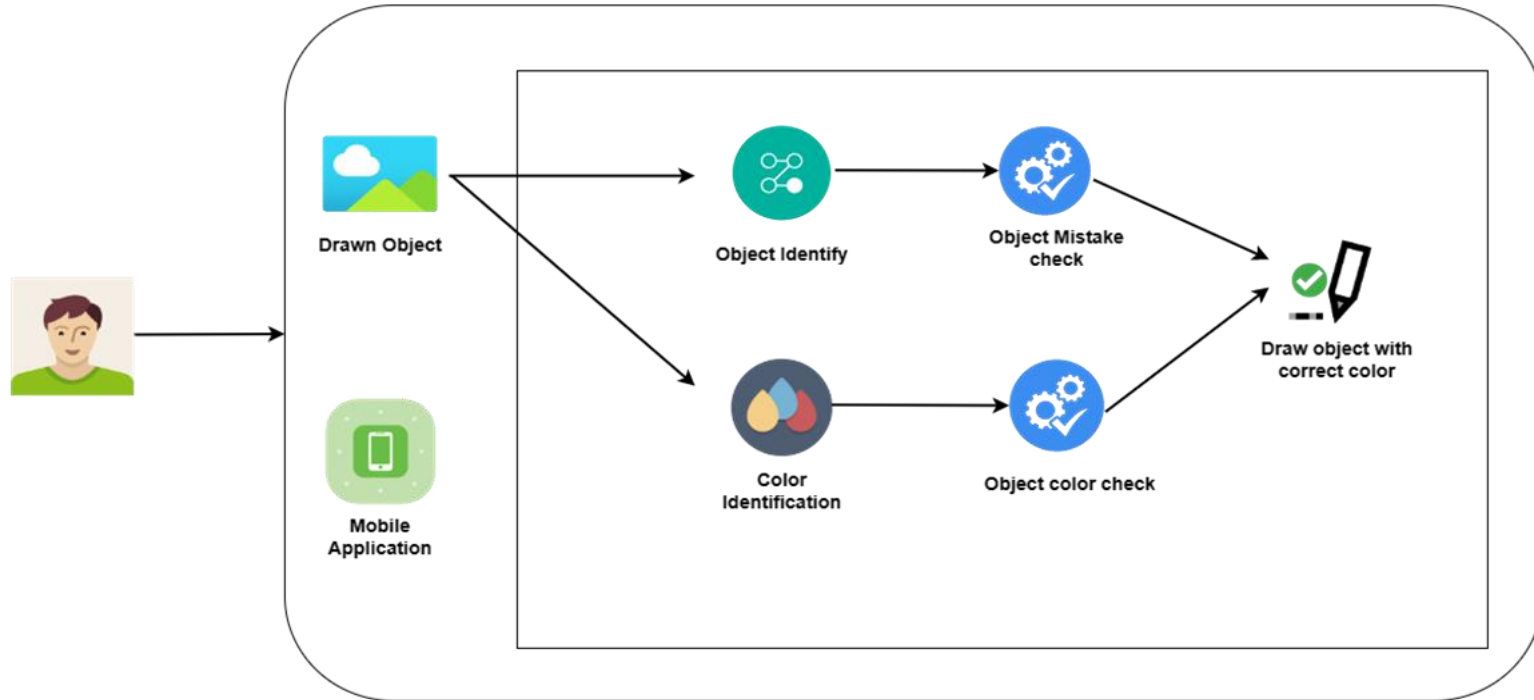
**50%**

## Expected Progress

- Develop frontend Application
- Connect backend and frontend
- Identify color mistakes done by student

- Automatically draw objects and auto apply colors.

# Hand drawn object detection with color identification





# 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



## How We Can Promote This App ?

- Facebook Advertisement
- Referral Program
- Notice
- TV Commercial



# How This App Make Money?

- 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**

