



Proposal Presentation

SignMath

Application to teach the Mathematics for hearing-impaired students
in primary schools



Team Members

- Supervisor:
- Co Supervisor:

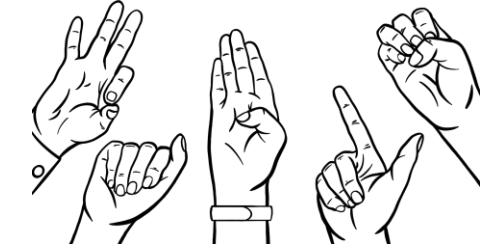


Student Name	Student Id
D.M.S.M.Dissanayaka	IT2014552
H.K.D.P.P.Weerarathne	IT19203072

Outline

- Introduction
- Background
- Research Problem
- High Overview Diagram
- Commercialization
- Budget
- Risk Mitigation
- Mobile App vs Web Application



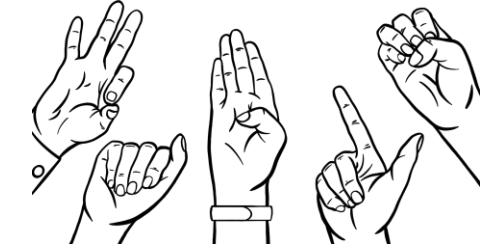


INTRODUCTION

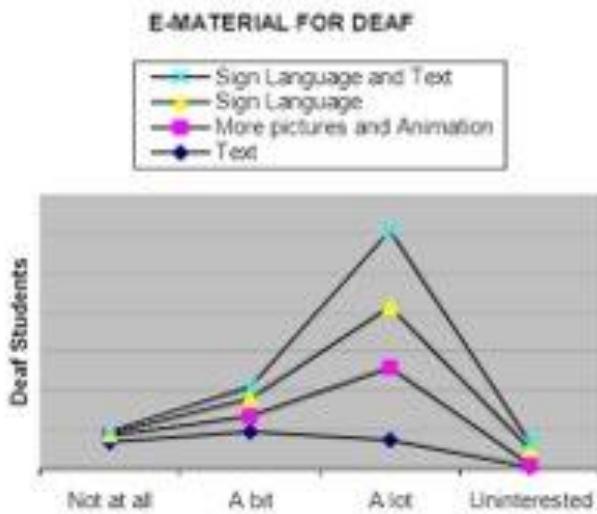
What is SignMath ?



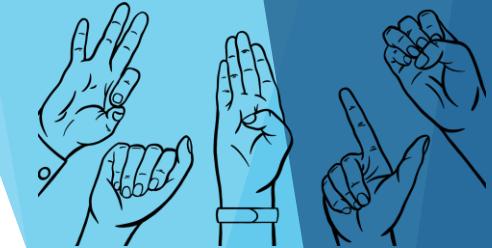
- ❖ Mathematics education for deaf children in Sri Lanka faces significant challenges due to language barriers and limited literacy.
- ❖ The majority of deaf children rely on sign language for communication, making it difficult for them to connect written words and signs with Sinhala language.
- ❖ These challenges need to be addressed in primary education to provide equal opportunities for deaf children in mathematics learning.

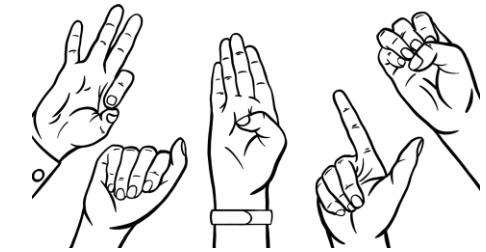


RESEARCH PROBLEM



- ❖ There were 1.2% deaf students (Age 1-14), approximately 5,390 registered deaf students in Sri Lanka as of the end of 2019
- ❖ 1/3 of sign language students tend to drop out of school during secondary education because they cannot pass main subjects like mathematics .
- ❖ The lack of understanding of the basic of Mathematical concepts
- ❖ Inability to express their knowledge and Emotions in the natural languages, especially in Sinhala.
- ❖ Teachers may not have the necessary training or resources to support deaf children in their mathematical education.





RESEARCH OBJECTIVE

Main Objective

Create a **Web Application to teach Mathematical Concepts to hearing-impaired students in primary schools** who are doing their studies using sign language.

Components

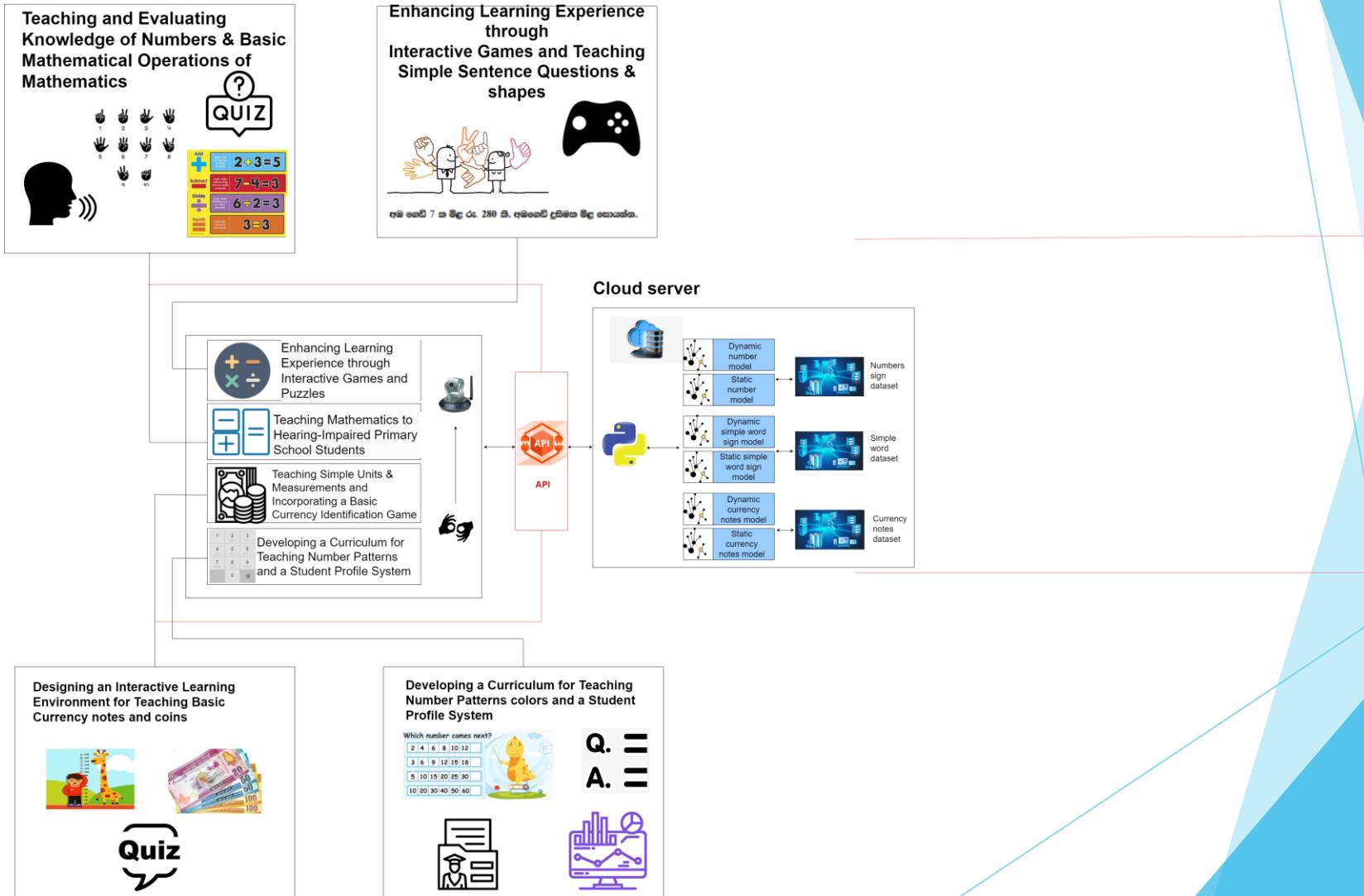
Teaching and Evaluating Numbers & Basic Mathematical Operations of Mathematics

Design Environment to Teaching Simple Sentence Questions and Shapes

Developing a Curriculum for Teaching Number Patterns, Colors and Evaluate Their Progress and Implementing a Student Profile System

Designing an Interactive Learning Environment for Teaching Basic Currency Notes and Coins

High Overview Diagram



Data Collecting



The School for the deaf (Rathmalana)

Pizza Hut New Branch (Thimbirigasyaya)

National Institute of Education (NIE -
Maharagama)



The School for the deaf (Ratmalana)



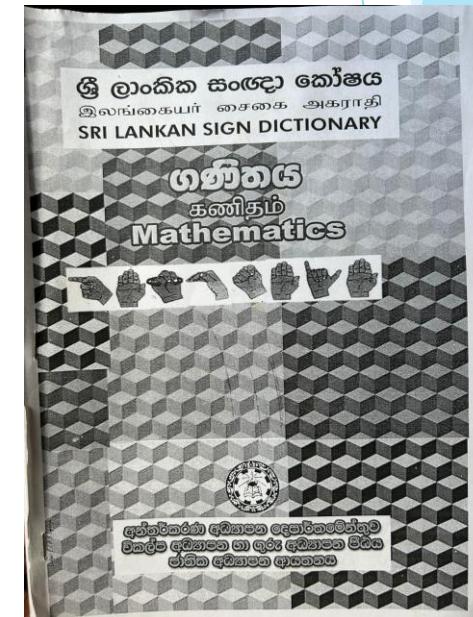
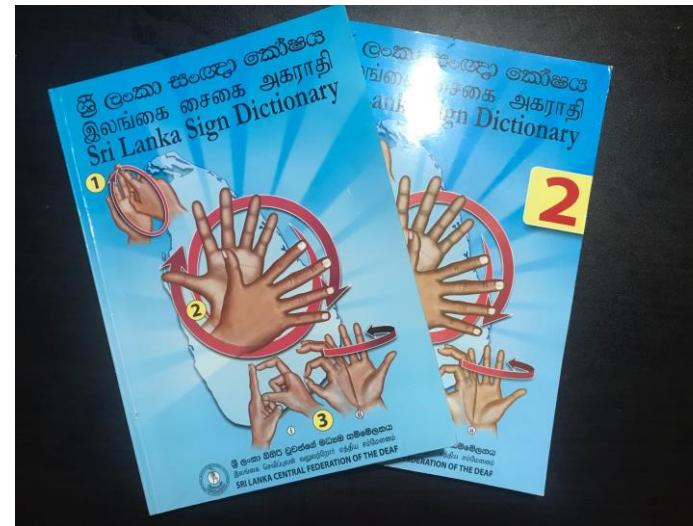


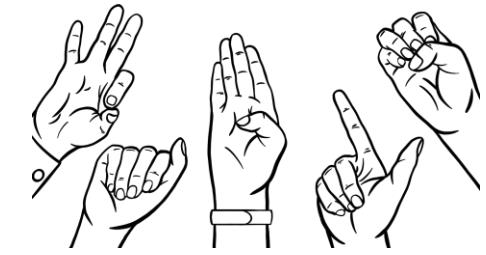
Pizza Hut New Branch (Thimbirigasyaya)



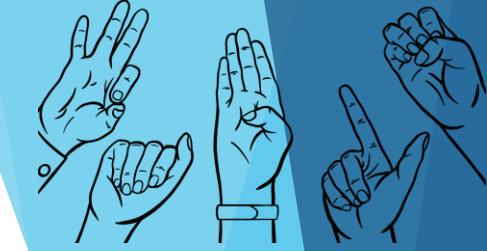


National Institute of Education (NIE – Maharagama)

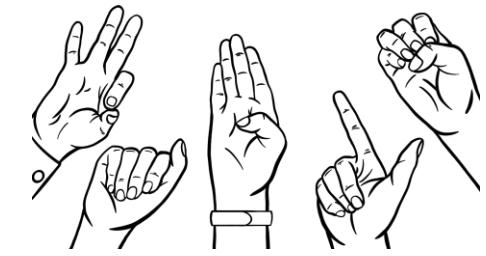




COMMERCIALIZATION

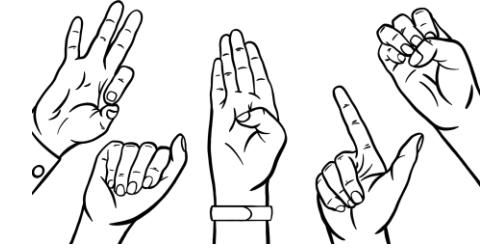


- ❖ For National Institute of Education – Rs. 2000 per month
(Training Teachers)
- ❖ Registration Fee– Rs.1000
- ❖ Child schools Free Subscription (Only Registration Fee)– No Charge
- ❖ For Deaf Training Center – Rs. 3000 per month

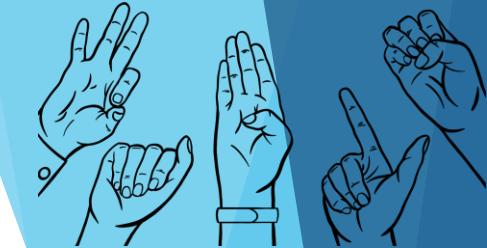


Budget

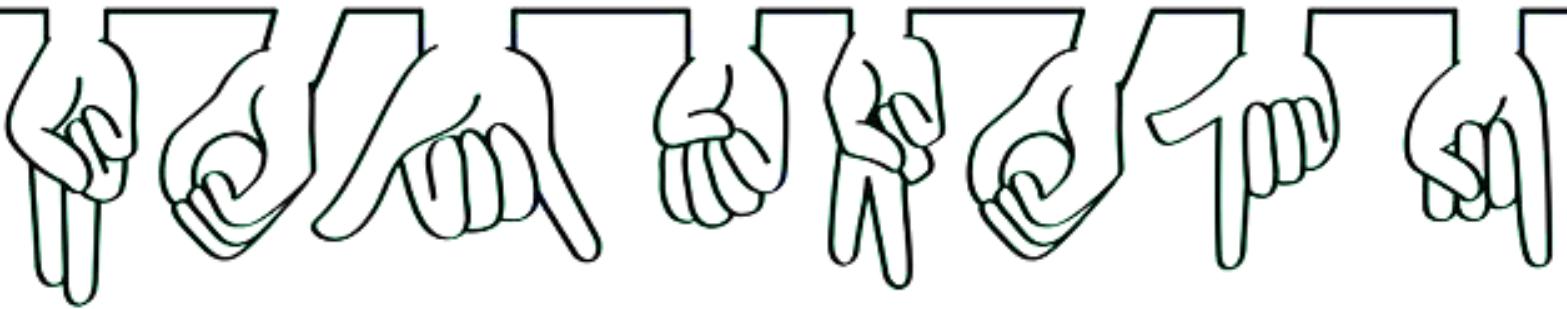
RESOURCES	PRICE
Domain & Server Cost	12,000
Traveling cost	4,000
Internet	3,000
Documentation & Printing Cost	3,000
Resources Books	1,200
Stationary	1,000
Total	24,200



RISK MITIGATION



Risk	Solution
No proper datasets available in websites(Sinhala)	Create dataset manually
Some sign models has less record of amounts	Remove those sign models
Algorithms sometimes failed to identify demonstration	Improve the size of dataset
Background issues	Keep the empty backgrounds



IT20145552 | D.M.S.M.Dissanayaka

BSc(Hons)Information Technology Specializing in Information Technology

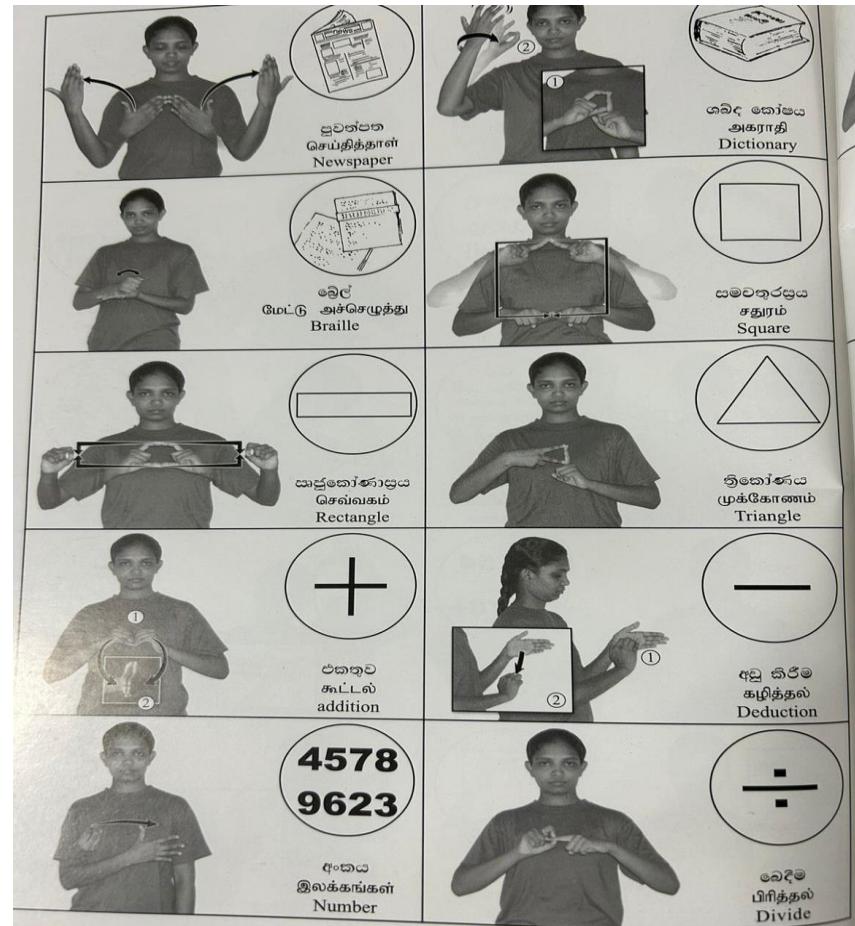
Designing a Novel Interactive Learning Environment for Enhancing Learning Experience through Shapes Comprehension

Introduction

- ▶ Majority of Sri Lankan's mother language is **Sinhala** and most of Sri Lankan kids struggles with understanding of combination of signs.
- ▶ Teaching shapes to Deaf and Mute Students is challenging due to the lack of interaction and inclusive learning tools. This research focuses on designing a novel e-learning environment that combines
 - Sign Language,**
 - Visual Aids** to enhance shape comprehension.
- ▶ The aim is to create an attractive **Sinhala usage platform** that will help these students in more effective and independent learning shapes.

Background

- ▶ The two types of signs in teaching shapes for deaf and mute students are:
 - ▶ Static Signs
 - ▶ Dynamic Signs



Research Problem

► Problem:

- ▶ How can an active learning environment be created through Sinhala Platform (Sinhala Language) to improve shape comprehension for deaf and dumb students through sign language, visual aids and interactive features???

► Answer:

- ▶ To address this problem, an innovative learning platform will be developed that integrates:
 1. Sign Language Integration
 2. Visual and Interactive Features
 3. Personalized Learning
 4. Continuous Practice Tools

Objectives

Main Objectives

Implement a system that Enhancing Learning Experience through Assessment and Design Environment to Teaching Simple Sentence mathematics And simple Shapes.



Create lesson plans and Teaching signs of simple Sentences math's.

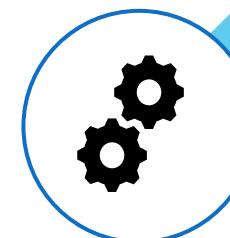
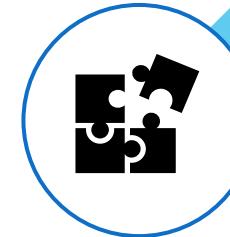


Image Pre-Processing and training the model, Creating a Dataset



Training the model and Generating quizzes



Checking students' physical demonstrations and answer evolution

Research Gap

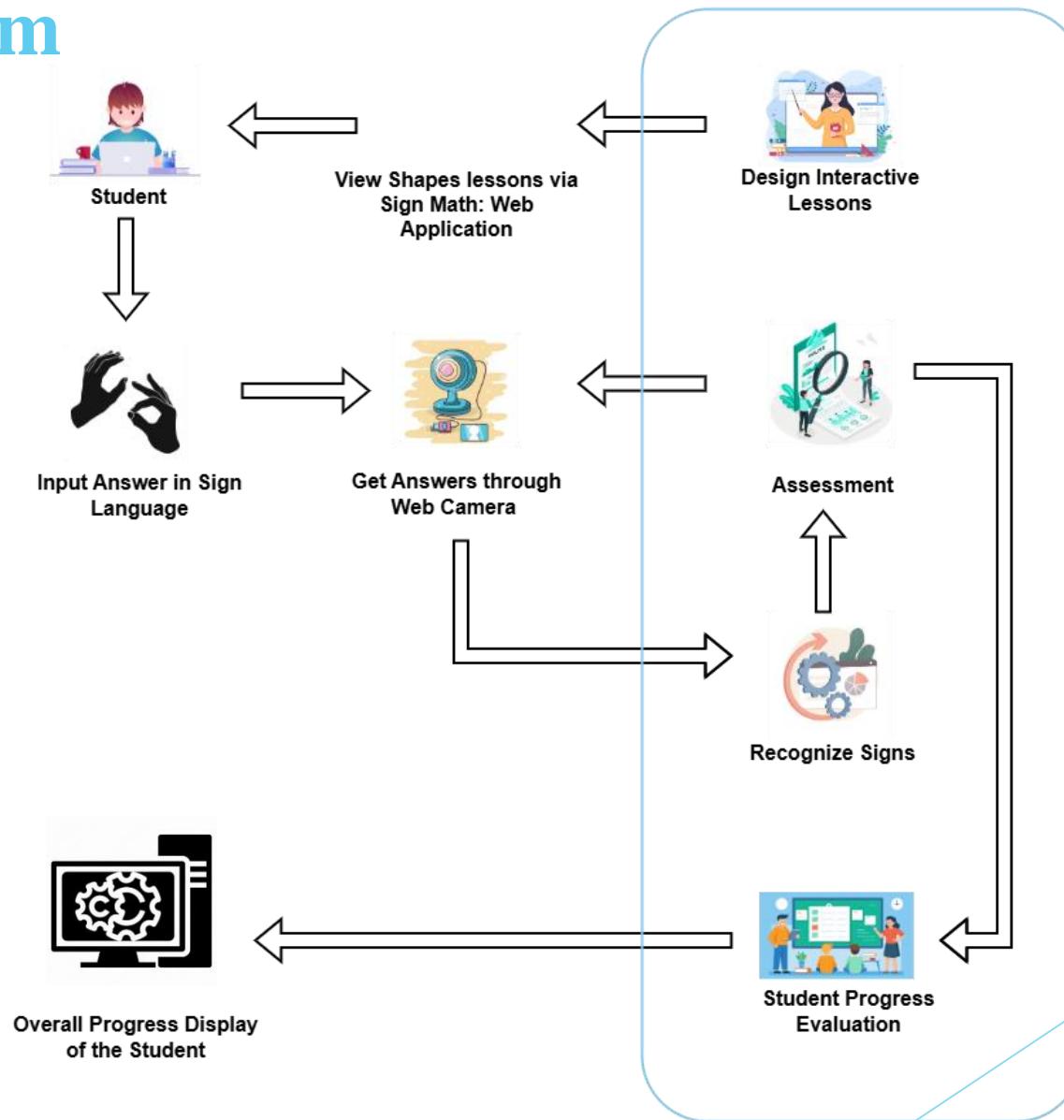
- ▶ In existing system,
 - ▶ No Sinhala sentences mathematics Teaching Facility only ASL
 - ▶ User cannot create sign photos (limited)
 - ▶ No Realtime Automated
 - ▶ Specially Not Designed for



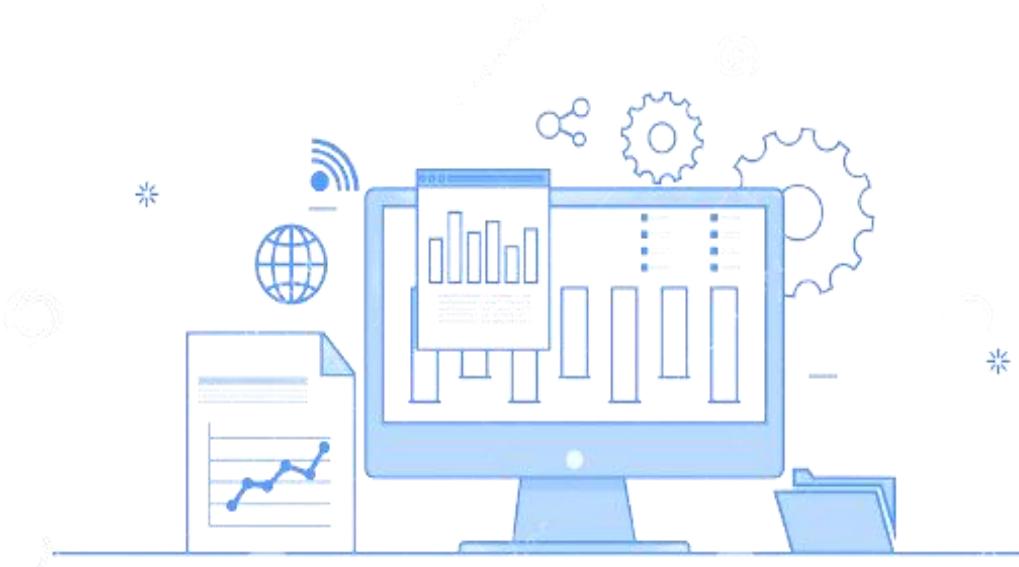
Research Gap

System Features	Math Signs Game	ASL Kids	MathSigner	DragonBox	SignSchool	Proposed Solution
Interactive learning platform Sinhala as sentences	✗	✗	✗	✗	✗	✓
Sign language Integration	✗	✗	✗	✗	✗	✓
Include compress method	✗	✗	✗	✗	✓	✓
Real time image capturing and give feedback	✗	✗	✓	✓	✗	✓
Assessment and generate report	✗	✓	✗	✗	✗	✓

System Diagram



SUB-OBJECTIVES COMPLETED Image Pre-Processing and Training The Model



Dataset

- ▶ Created a dataset with,
 - Different backgrounds
 - Slight variations in the shapes
 - Close to 100 hundred images for one shape

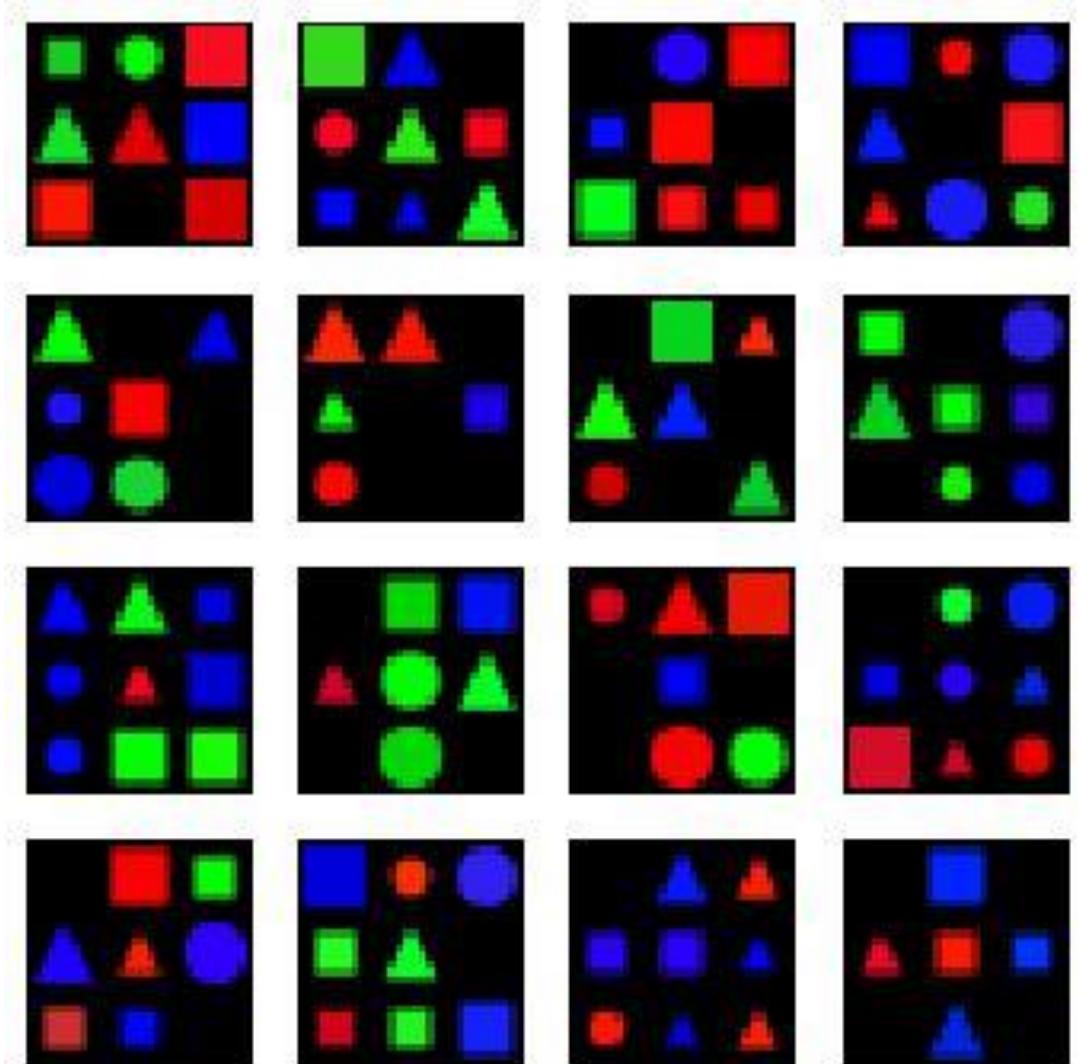
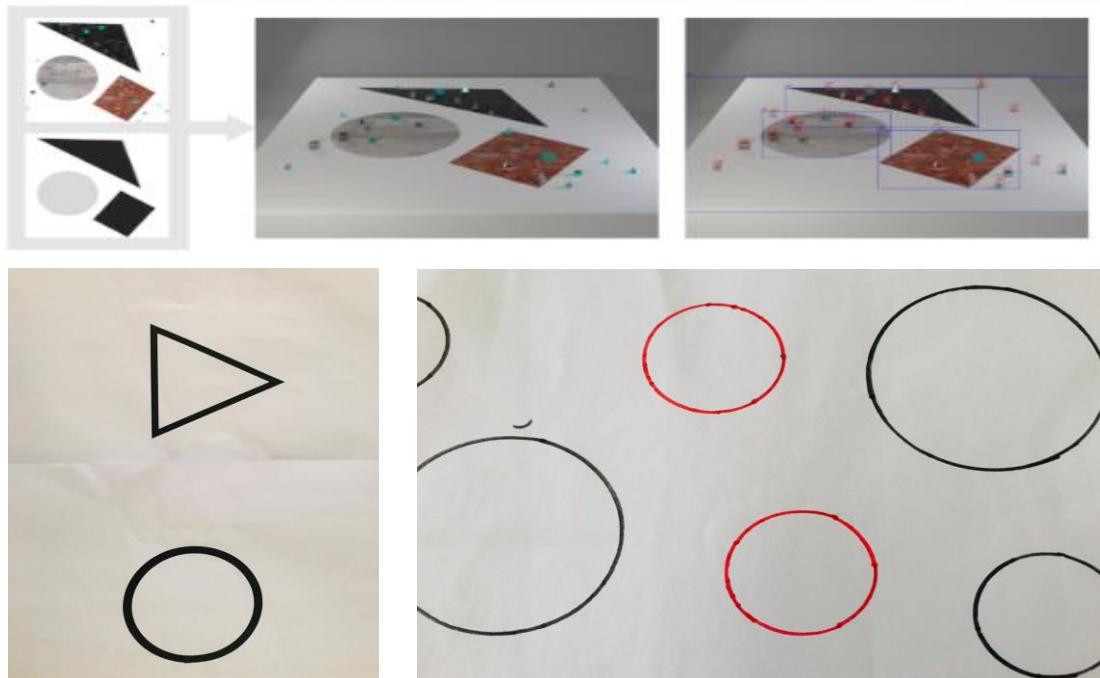
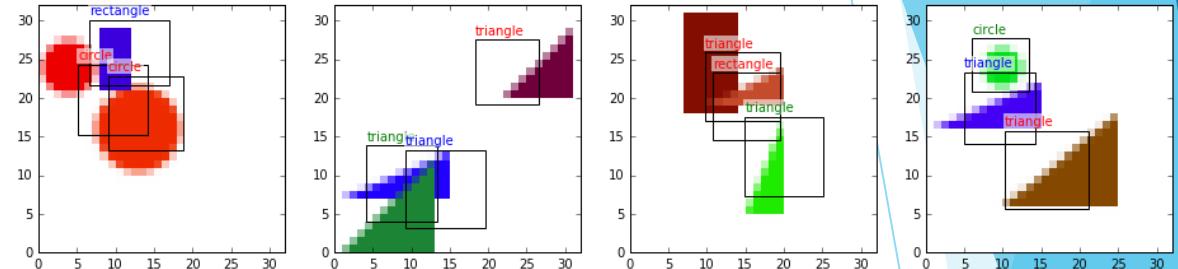


Image Pre-Processing

without pre-processing

High fluctuation precision-recall graphs
Less Map value
Hight train and validation loss
Without Gray scale



with pre-processing

Less fluctuation precision-recall graphs
High Map value
Less train and validation loss
With Gray scale

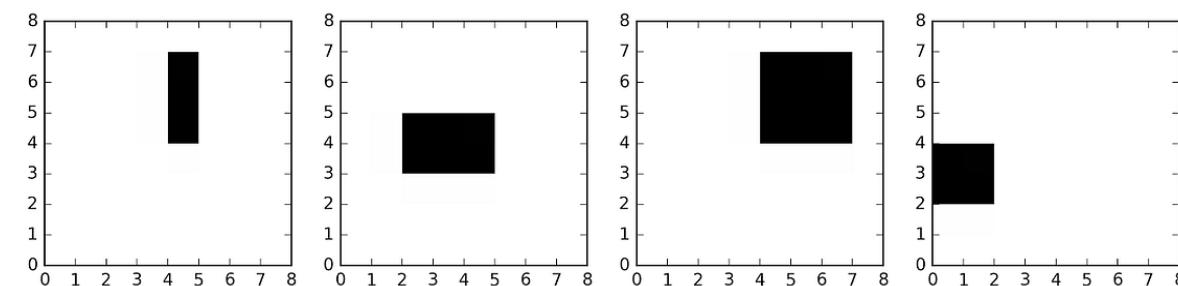
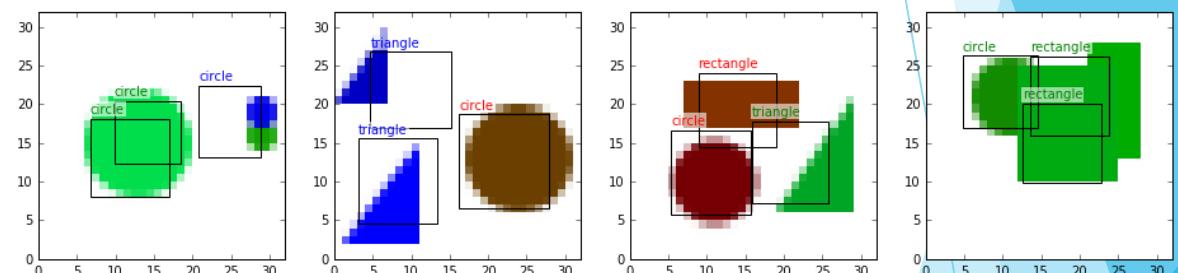
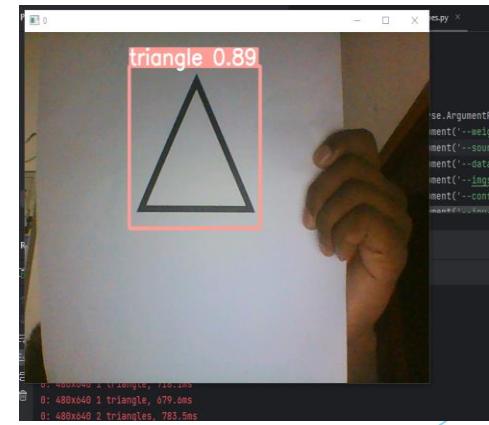
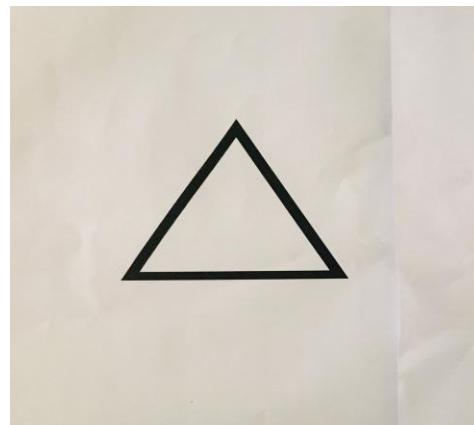
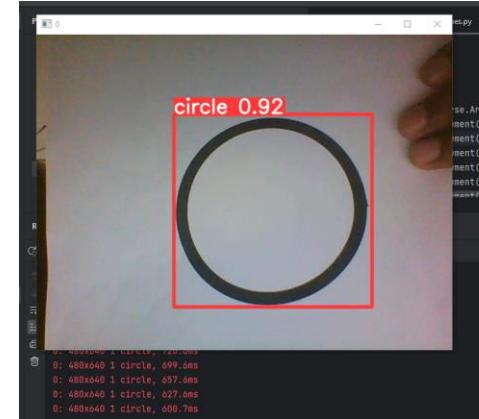
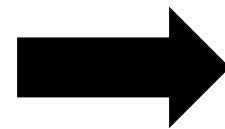


Image Pre-Processing



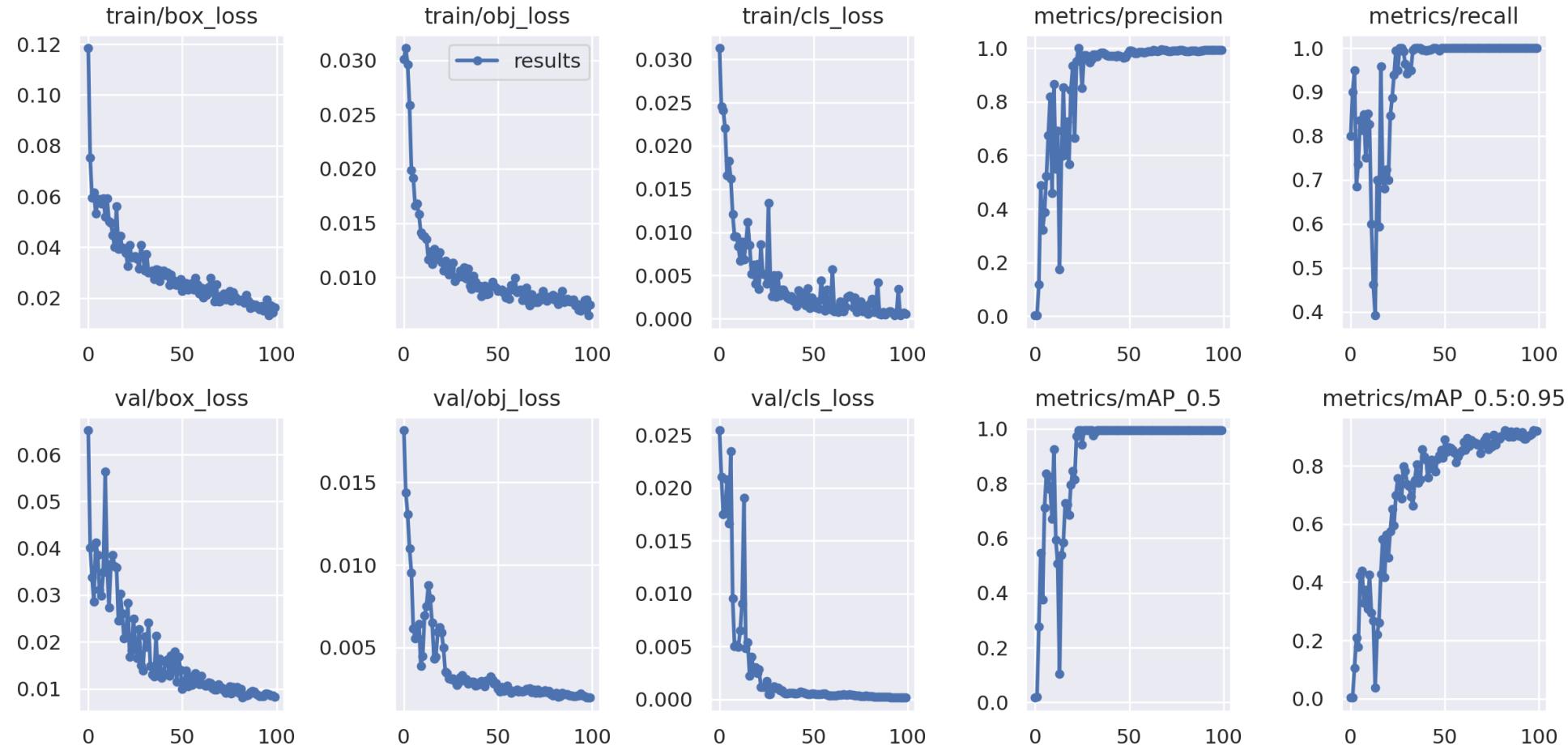
Training the Model

- Model trained using Google Colab online tool

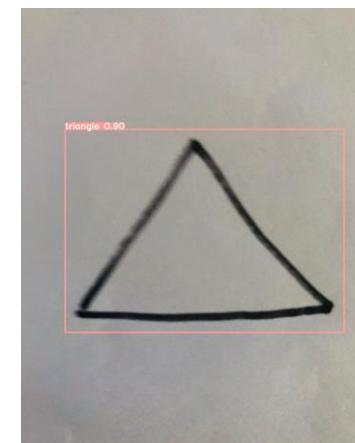
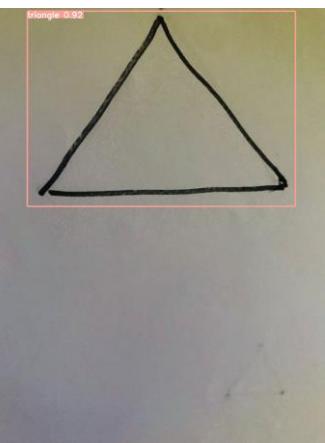
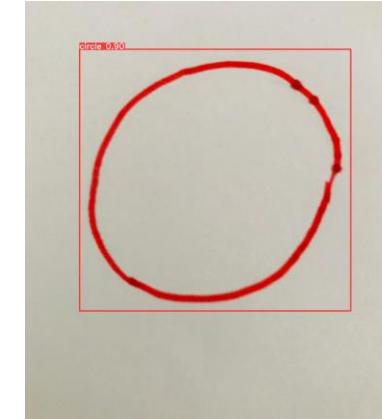
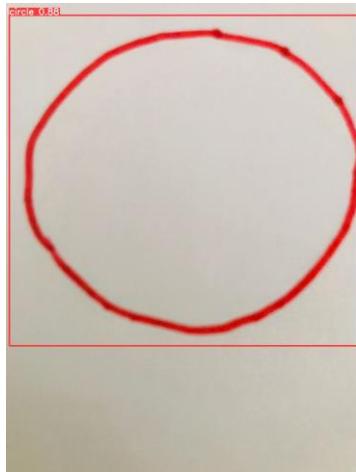
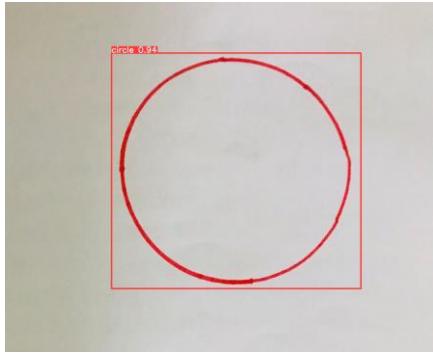


```
100 epochs completed in 0.096 hours.  
Optimizer stripped from runs/train/exp/weights/last.pt, 14.4MB  
Optimizer stripped from runs/train/exp/weights/best.pt, 14.4MB  
  
Validating runs/train/exp/weights/best.pt...  
Fusing layers...  
Model summary: 157 layers, 7015519 parameters, 0 gradients, 15.8 GFLOPs  
      Class    Images  Instances       P        R     mAP50   mAP50-95: 100% 1/1  
        all       20       20    0.988      1    0.995    0.922  
      circle      20       10    0.981      1    0.995    0.96  
      triangle     20       10    0.995      1    0.995    0.885  
Results saved to runs/train/exp
```

Training the Model

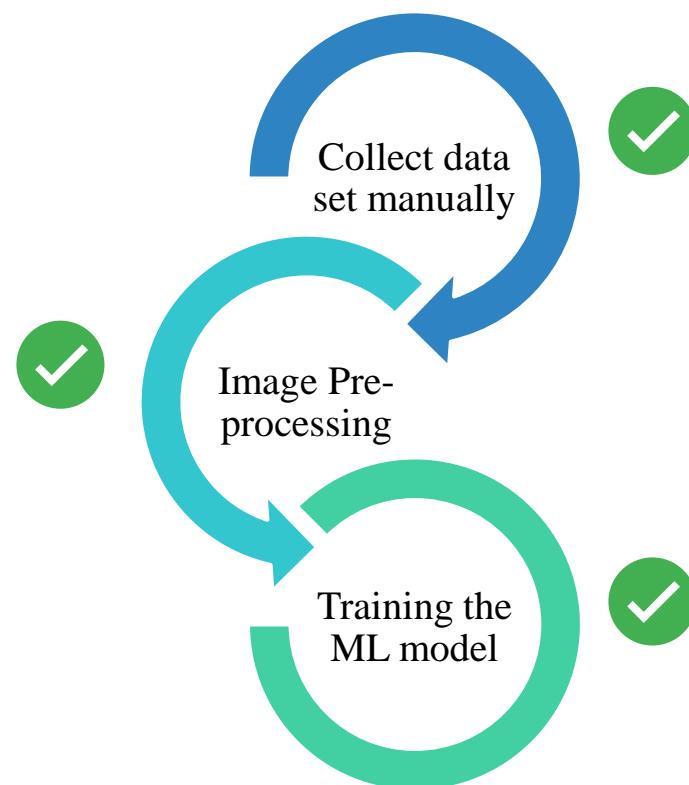


Accuracy Check

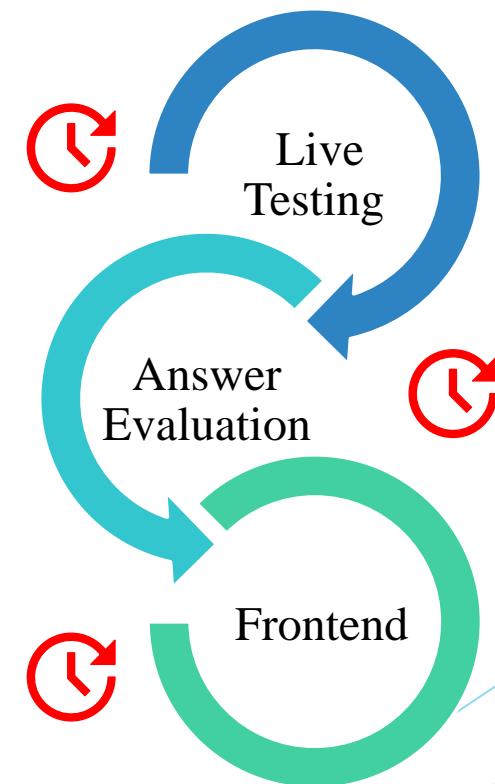


Completion and Future Works

Completed Implementations



Future Implementations



Used Technologies

- ▶ Keras (implement neural networks)
- ▶ React (Frontend)
- ▶ Firebase (Database)
- ▶ Google Colab (Training the model)
- ▶ Python(model)
- ▶ REST API
- ▶ Python flask (creating rest API)
- ▶ Open CV(image processing)
- ▶ Tensor Flow

 TensorFlow



 Keras

 Firebase

 + colab
python



Requirements

FUNCTIONAL REQUIREMENTS

- ✓ Sign Language Integration
- ✓ Word Identifying Quizzes
- ✓ Answer Evaluation & score

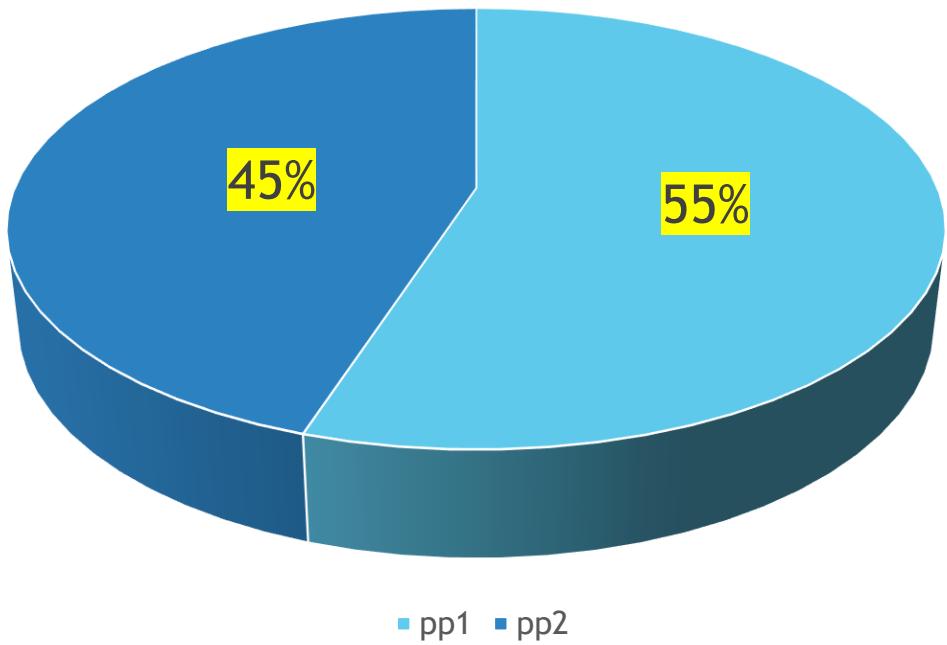
NON-FUNCTIONAL REQUIREMENTS

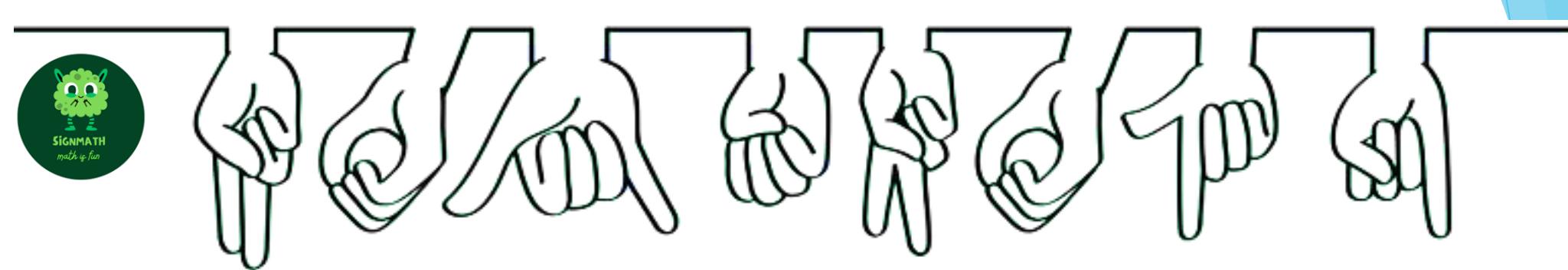
- ✓ Usability
- ✓ Reliability
- ✓ Portability
- ✓ Speed

USER REQUIREMENTS

- ✓ Laptop
- ✓ Internet Connection
- ✓ Front Camera

progress





IT19203072 | H.K.D.P.P.Weerarathne

BSc(Hons)Information Technology Specializing in Information Technology



Designing a Novel Interactive Learning Environment for Teaching and Evaluating Numbers & Basic Mathematical Operations of Mathematic

Introduction

Design an interactive learning environment for teaching Sinhala Alphabet and vowel signs

- Background
- Research Problem
- Main Objectives and Sub Objectives



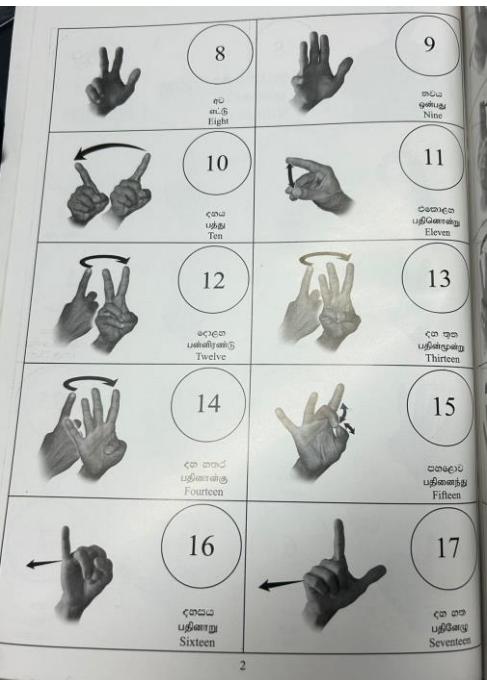
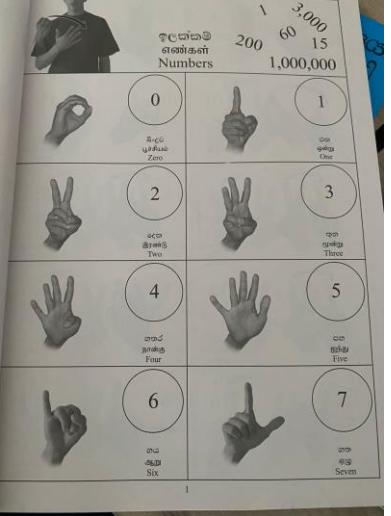
Background

Why Mathematical Basic Concepts Important to Deaf Children?

What type of Quizzes Can We Provide Through the Web App ?

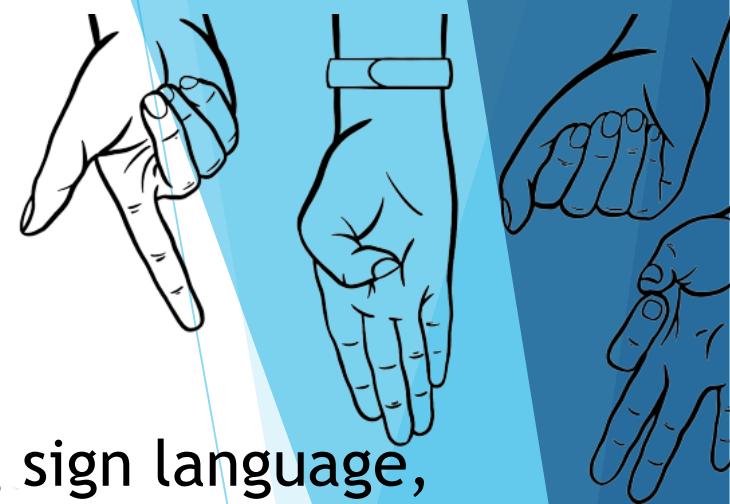
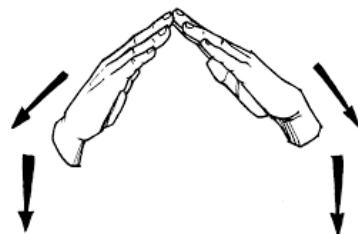
Background

There are 2 types of signs in the in Number



Research problem

- ❖ How can an interactive learning environment incorporating sign language, visual aids, and machine learning technologies be designed?
- ❖ Can this novel approach improve the numeracy skills of deaf and mute children?
- ❖ Most of the time they use common (universal) signs in sign language to communicate. Therefore, they don't get an opportunity to revise their knowledge outside of the classroom.



Research Gap

System Features	Math Signs Game	ASL Kids	MathSigner	DragonBox	SignSchool	Proposed Solution
Teaching Numbers in Sri Lankan Sign Language (SLSL)	✗	✗	✗	✗	✗	✓
Teaching Mathematics Basic Operations in Sri Lankan Sign Language (SLSL)	✗	✗	✗	✗	✗	✓
Generate Quizes	✓	✗	✗	✓	✗	✓
Student Evaluation Profile	✓	✗	✗	✗	✓	✓
Use of Machine Learning	✗	✗	✗	✗	✗	✓

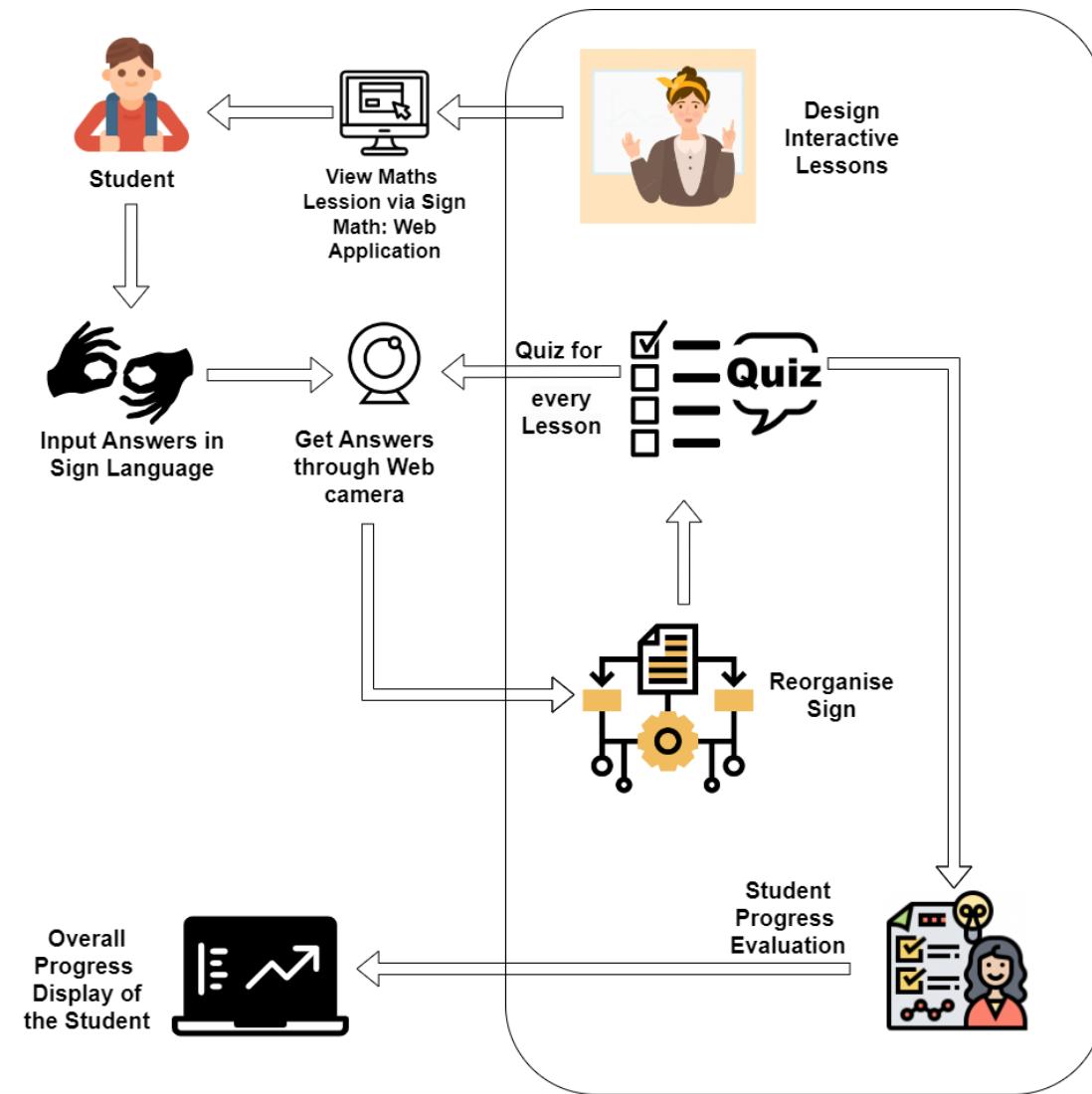
Specific Objectives & Sub Objectives

Main objective- Implement a system that teach and evaluate the understanding of numbers and basic mathematic operations to deaf and mute children

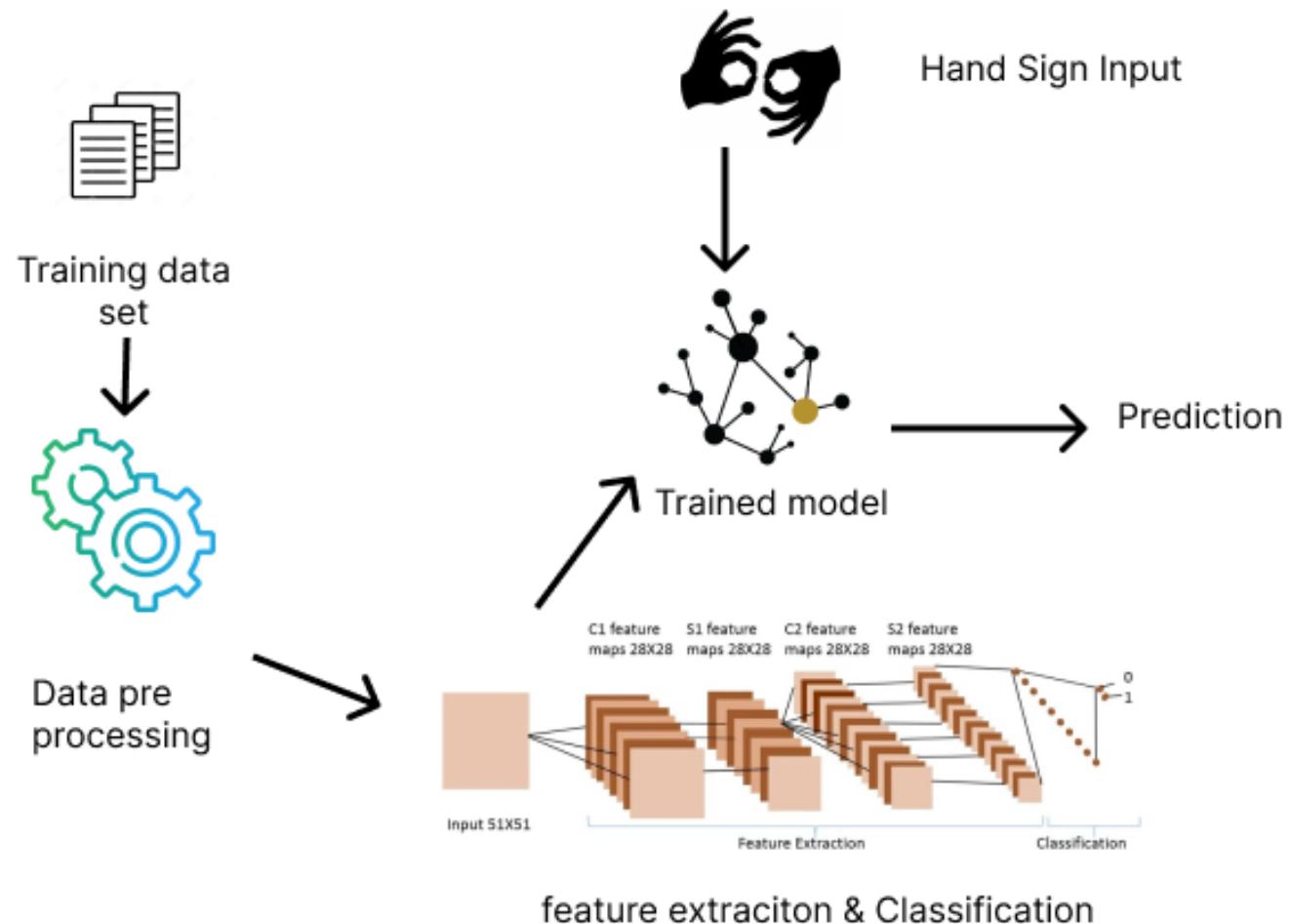
Sub Objectives

- Creating interactive lessons to teach Numbers and Operations (images, animations, videos).
- Grading Student's knowledge using quizzes.
- Evaluating students' overall progress.
- Identify finger spelling through the camera
 - Creating a Dataset
 - Building a separate model to Identify signed numbers
 - Image pre-processing
 - Training the model
 - Hands segmentation and tracking
 - Feature extraction and classification

System Overview Diagram



How finger Spelling is identified



Methodology

Individual lesson & quiz for each Number & Operation



QUIZ

Lesson interface(related sign+ Sinhala word +lip movement video)

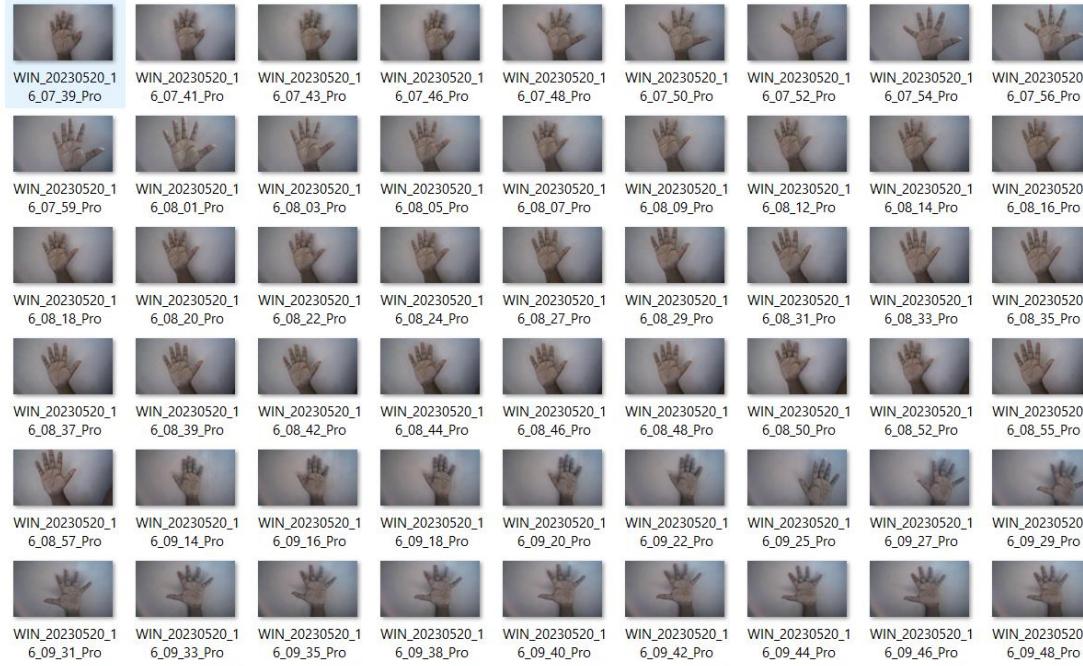
Show Sign to the Camera



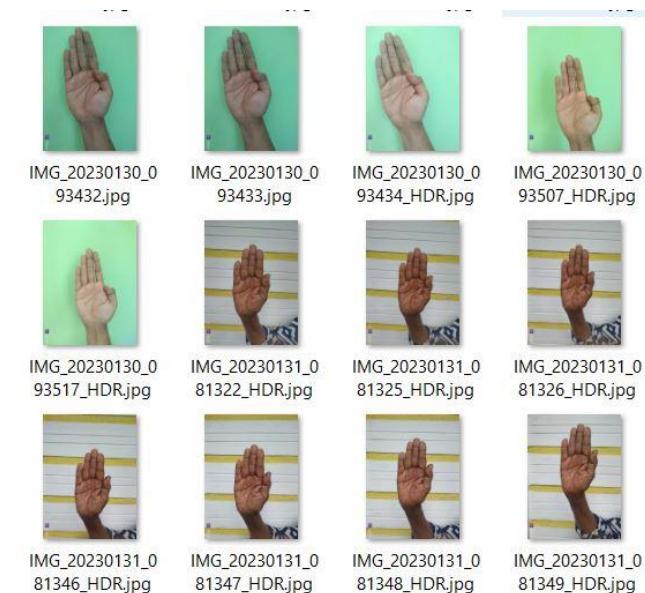
Quiz interface

Dataset

Created a Dataset with,



- Different backgrounds
- Hand signs from two persons
- Slight variations in the hand
- Close to 400 hundred images for one sign





Training the model

```
Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7026307 parameters, 0 gradients, 15.8 GFLOPs
  Class   Images  Instances      P      R    mAP50    mAP50-95: 100% 3/3 [00:01<
    all      66       66    0.912    0.963    0.995     0.86
    0       66       11    0.98      1        0.995     0.911
    1       66       11        1    0.967    0.995     0.847
    2       66       11    0.741      1        0.995     0.869
    3       66       11    0.754      1        0.995     0.851
    4       66       11        1    0.862    0.995     0.834
    5       66       11        1    0.952    0.995     0.849
Results saved to runs/train/exp

[Progress bar]

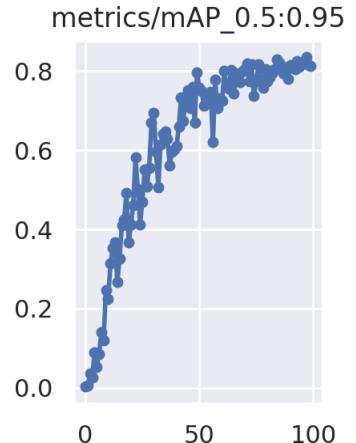
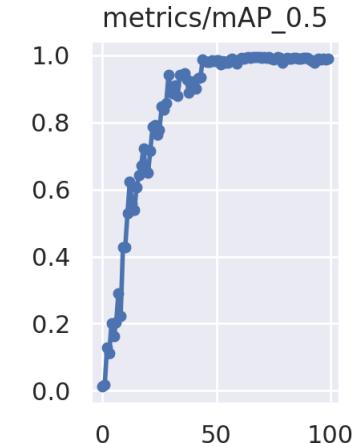
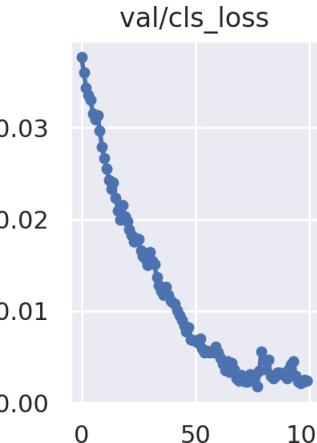
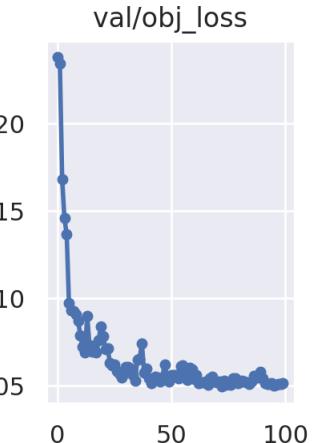
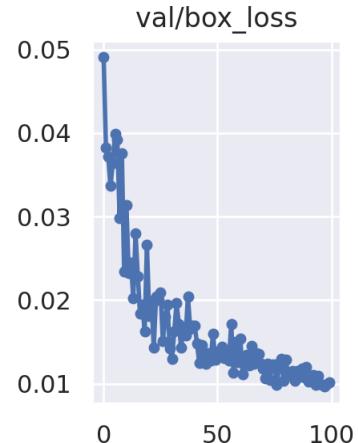
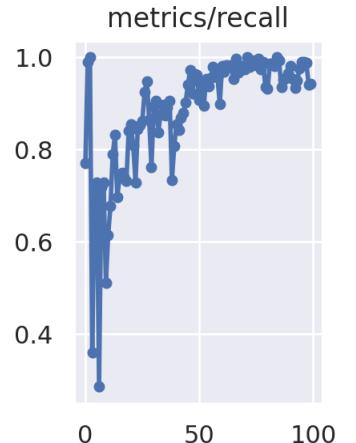
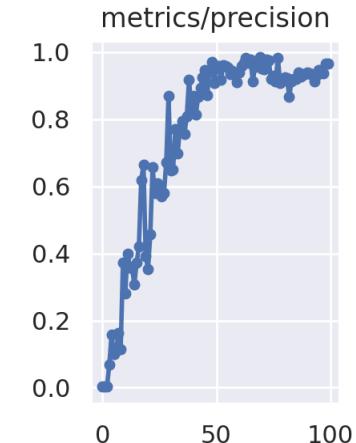
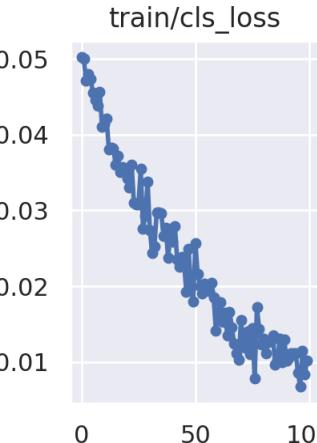
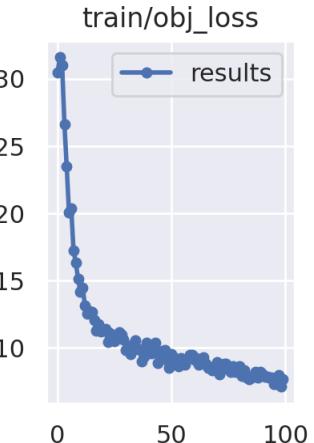
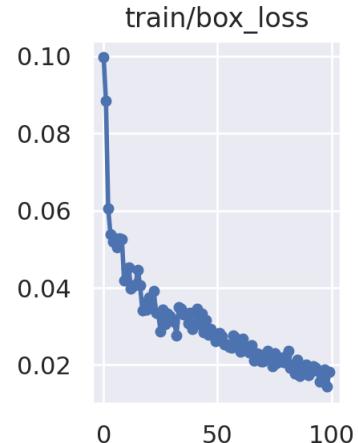
[Logs]
  ✓ 12m 26s completed at 11:52PM
```

100 epochs completed in 0.132 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 14.4MB
Optimizer stripped from runs/train/exp/weights/best.pt, 14.4MB

```
Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7026307 parameters, 0 gradients, 15.8 GFLOPs
  Class   Images  Instances      P      R    mAP50    mAP50-95: 100%
    all      96       96    0.937    0.987    0.989     0.835
    0       96       16    0.993    0.938    0.971     0.701
    1       96       16    0.943      1        0.995     0.816
    2       96       16    0.953      1        0.995     0.89
    3       96       16    0.974      1        0.995     0.844
    4       96       16    0.759      1        0.984     0.846
    5       96       16        1    0.984    0.995     0.915
Results saved to runs/train/exp
```

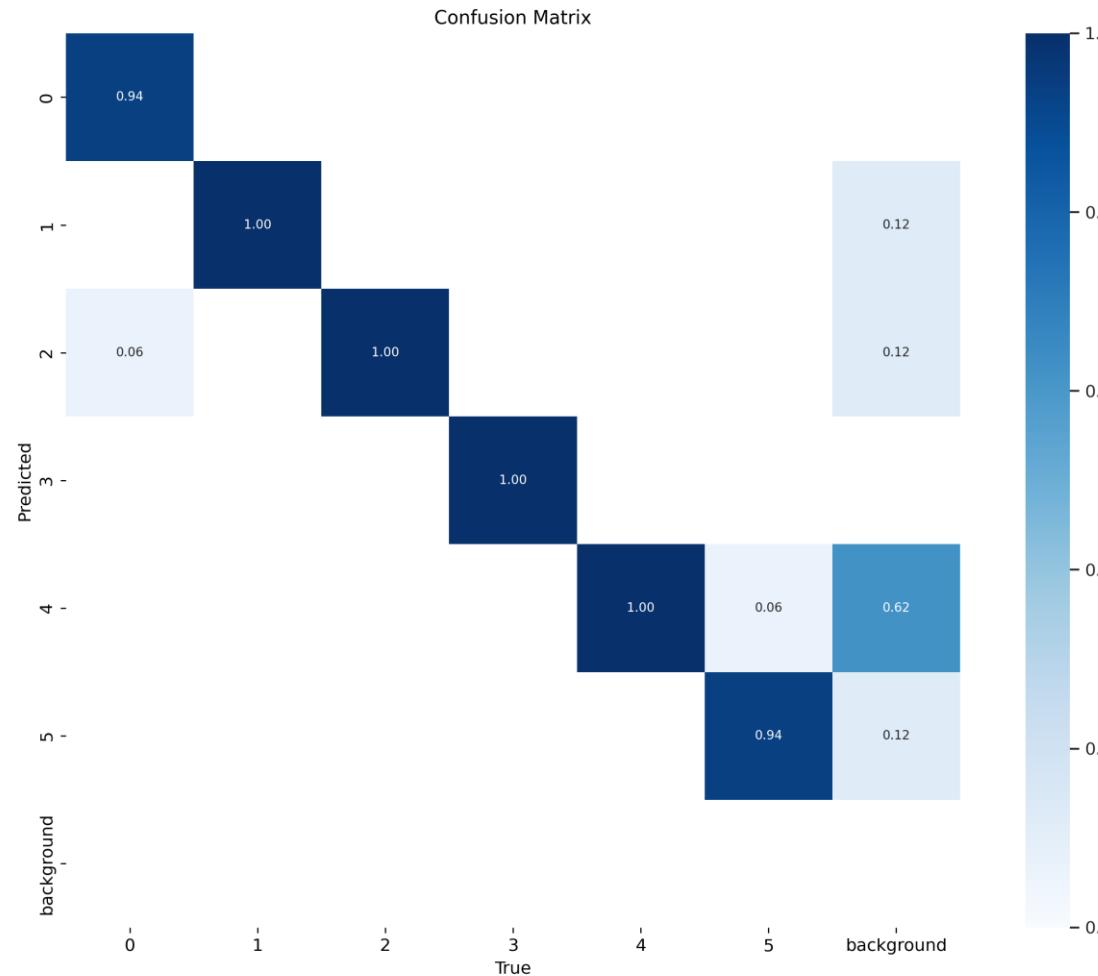
- Model trained using Google Colab online tool
- Separate dataset is used to validate the model

Accuracy Report

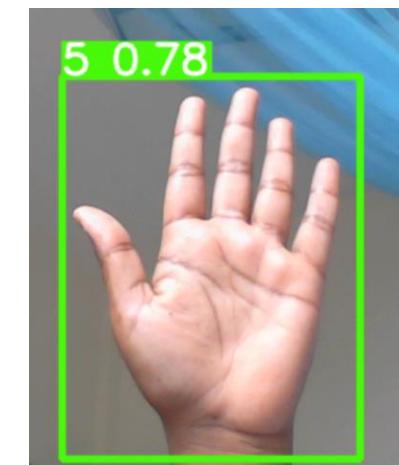




Accuracy Reports



Accuracy Check



Detecting Hand sign using camera(partial)

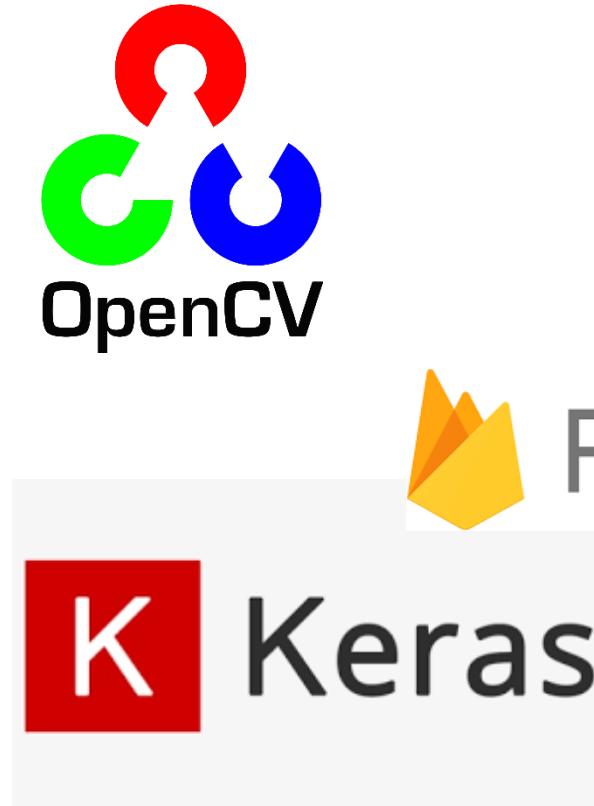
```
detect_numbers.py x
188     strip_optimizer(weights) # update model (to fix SourceChangeWarning)
189
190     print('Done')
191
192
193 def parse_opt():
194     parser = argparse.ArgumentParser()
195     parser.add_argument('--weights', nargs='+', type=str, default=ROOT / 'models/best_numbers.pt', help='model path or triton URL')
196     parser.add_argument('--source', type=str, default=ROOT / '0', help='file/dir/URL/glob/screen/0(webcam)') #0 webcam #data/images/numbers
197     parser.add_argument('--data', type=str, default=ROOT / 'data/coco128.yaml', help='(optional) dataset.yaml path')
198     parser.add_argument('--imgsz', '--img', '--img-size', nargs='+', type=int, default=[640], help='inference size h,w')
199     parser.add_argument('--conf-thres', type=float, default=0.25, help='confidence threshold')
200     parser.add_argument('--iou-thres', type=float, default=0.45, help='NMS IoU threshold')
201     parser.add_argument('--max-det', type=int, default=1000, help='maximum detections per image')
202     parser.add_argument('--device', default='', help='cuda device, i.e. 0 or 0,1,2,3 or cpu')
203     parser.add_argument('--view-img', action='store_true', help='show results')
204     parser.add_argument('--save-txt', action='store_true', help='save results to *.txt')
205     parser.add_argument('--save-conf', action='store_true', help='save confidences in --save-txt labels')
206     parser.add_argument('--save-crop', action='store_true', help='save cropped prediction boxes')
207     parser.add_argument('--nosave', action='store_true', help='do not save images/videos')
208     parser.add_argument('--classes', nargs='+', type=int, help='filter by class: --classes 0, or --classes 0 2 3')
209     parser.add_argument('--agnostic-nms', action='store_true', help='class-agnostic NMS')
210     parser.add_argument('--augment', action='store_true', help='augmented inference')
211     parser.add_argument('--visualize', action='store_true', help='visualize features')
212     parser.add_argument('--update', action='store_true', help='update all models')
213     parser.add_argument('--project', default=ROOT / 'runs/detect', help='save results to project/name')
214     parser.add_argument('--name', default='avn', help='save results to project/name')
...
run()
```

- Presence of the hand in the camera input is identified
- Image of the hand is used as an input for predicting the hand sign.
- Return the output to the web application
- Ex:



Technologies used

- ▶ Keras (implement neural networks)
- ▶ React (Frontend)
- ▶ Firebase (Database)
- ▶ Google Colab (Training the model)
- ▶ Python(model)
- ▶ REST API
- ▶ Python flask (creating rest API)
- ▶ Open CV(image processing)
- ▶ Tensor Flow



Requirements

- ▶ Functional Requirements
- User should be able to view individual lessons.
- Users should be able to measure their knowledge after the lesson by attempting the quizzes in a lesson
- Sign input should be identified using the image acquired through the camera.
- Quizzes should be graded.
- Individual students' progress should be evaluated by quiz marks and displayed.



Progress

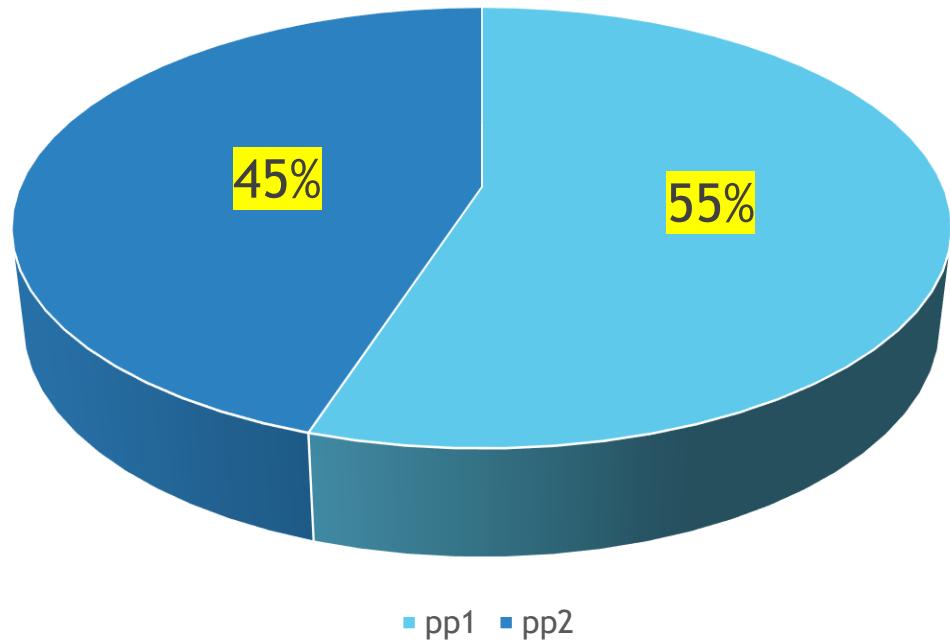
- ▶ A New dataset is created for Numbers in Sinhala sign language.
- ▶ Build the model to identify signs.
- ▶ Adjusting and training the model to achieve high accuracy
- ▶ Validating the model using the test dataset
- ▶ Sign Detection using camera input

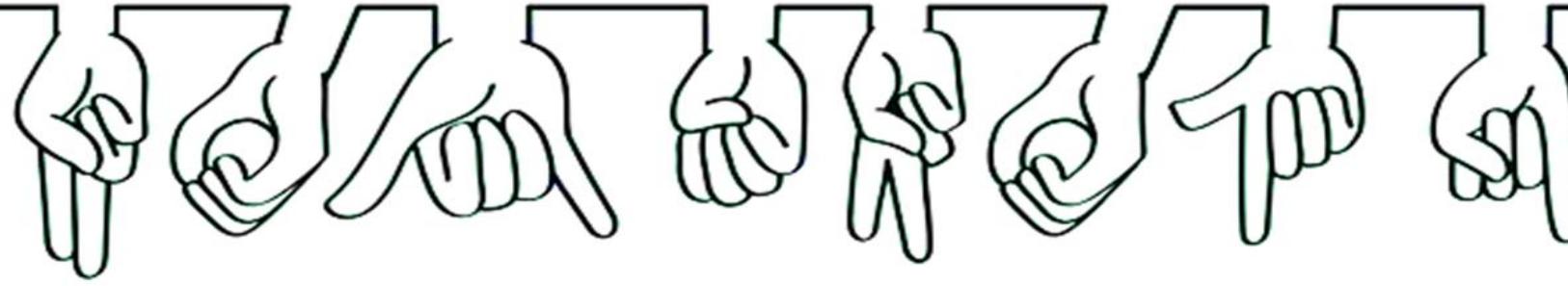


Future Implementation

- complete sign detection using camera input
- Designing interactive lessons to teach more numbers (front-end)
- Evaluating student's progress
 - Grading quizzes
 - Calculate overall student's progress

progress





BSc (Hons) in Information Technology Specializing in Computer System and Network Engineering

Designing an Interactive Learning Environment for Teaching Basic Currency Identification

Outline

- Introduction
- Background
- Research Problem
- Objectives
- Research Gap
- System Diagram
- Methodology



BACKGROUND

Why we need to teach currency notes and coins ?

- Use of currency notes and coins is essential nowadays.
- Majority of children struggled with understanding the combinations of signs
- Teachers also struggled to teach them in school
- That it is good to give them an understanding of currencies at an early age.

**WHY
Currency notes
and coins?**



RESEARCH PROBLEM

How to reduce the difficulties Of teachers while teaching ?

- Teach combined signs with user-friendly interface

- Create Sign Videos with Sentences

Teachers can demonstrate create their difficult part with sign videos-so they don't need to teach again and again

- Currency identifying Quizzes for related sign videos

- Currency identifying quizzes using Sinhala medium



OBJECTIVES

Main Objective

Designing an Interactive Learning Environment for Teaching Basic Currency Identification.

Sub-Objective

Teaching Sri Lanka Basic Currency notes and coins.

Generate Quizzes, Showing Answers to the camera.

Answer Evaluation & Correction.



Why sign language videos better than Images?

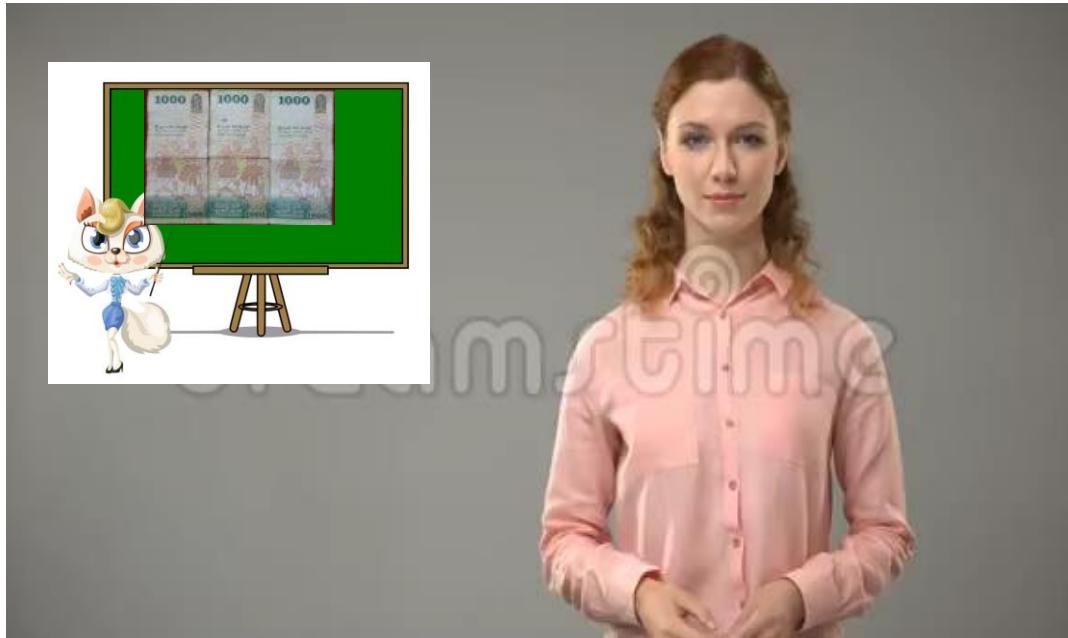
- The video makes learning convenient
- Videos have the power to express strong emotions that encourage viewers.
- Images can not contain movements, but videos can.
- Brain can also process visuals much quicker than it can text.
- Children likes to watch videos than Reading
- Enjoyable & memorable



According to studies, the average viewer remembers 95% of a message when it is watched.

How to increase the student's memorization of the Sign language Easily?

- Provide multiple choice questions to identify the correct sentence for the related video

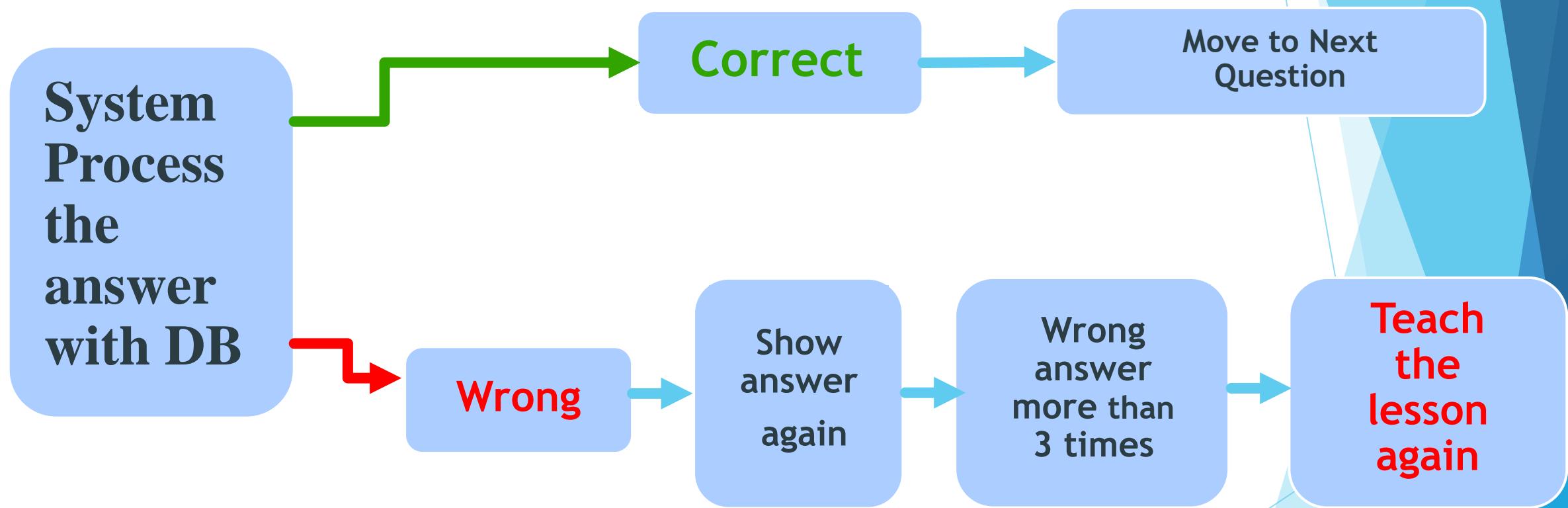


ರೂಪ್ಯಾಲ್ 100 ರೂ

ರೂಪ್ಯಾಲ್ 1000 ರೂ

ರೂಪ್ಯಾಲ್ 500ರೂ

How to provide Answer Evaluation for the students ?



RESEARCH GAP

In Existing systems,

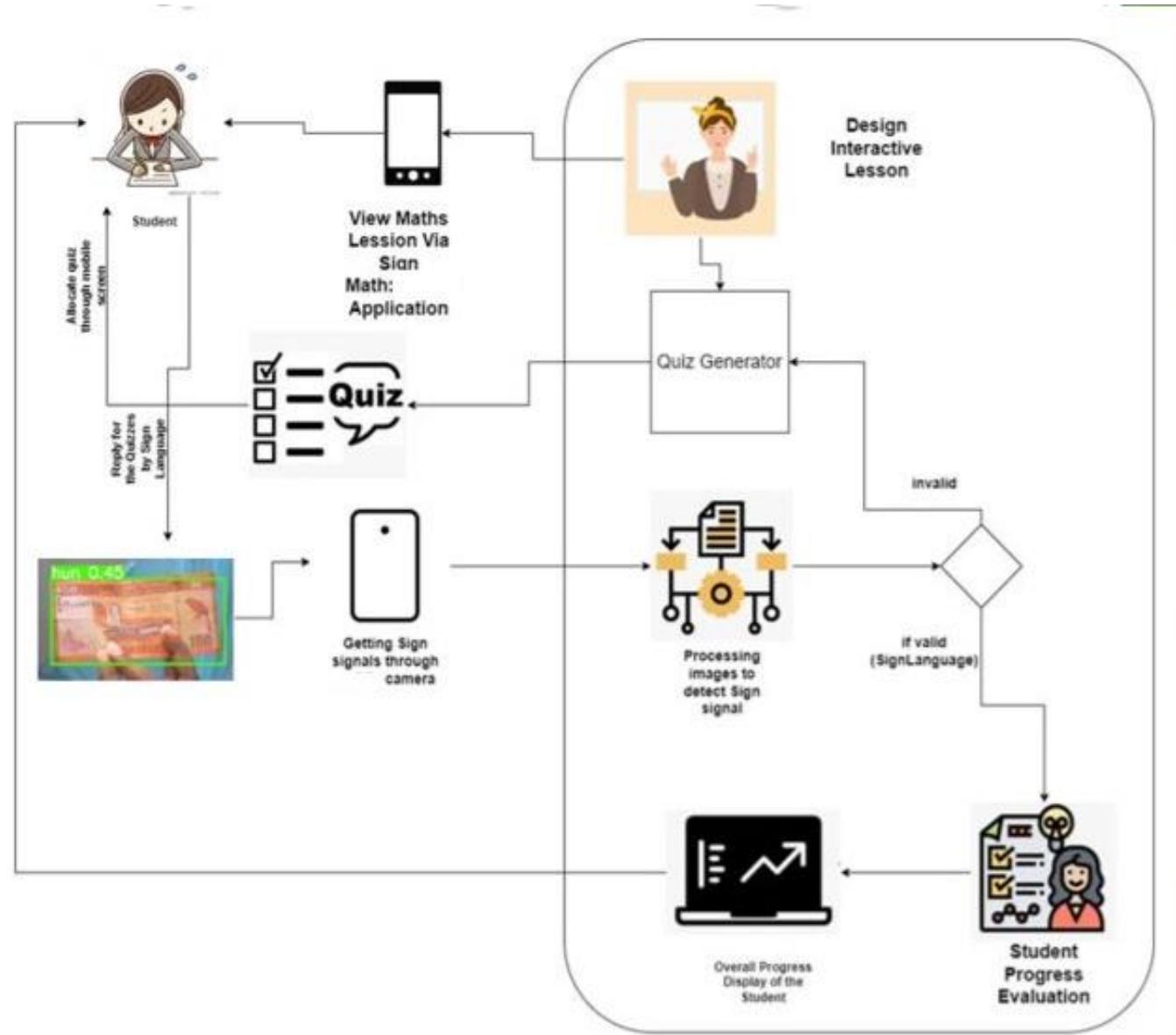
- Use Sri lanka Sign Language (SLSL) to teach maths.
- User can not create Sign videos (Data limited)
- Existence of descriptions, and quizzes in Sinhala medium.
- Designed for children
- Answer Evaluation & Correction



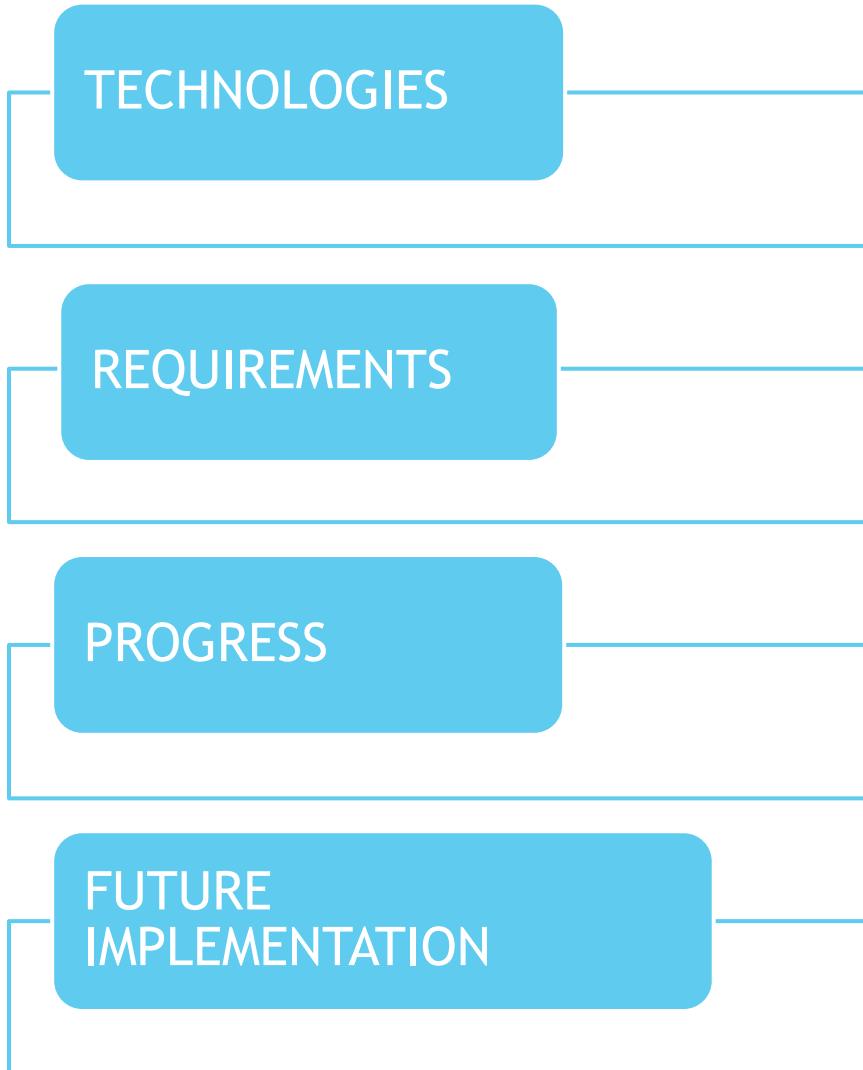
RESEARCH GAP

System Features	Mathwa	Mathsigner	DeafSign	Teach currency note for <u>disabled</u> students	DragonBox Numbers	Proposed Solution
Teaching currency notes in Sinhala	✗	✗	✗	✗	✗	✓
Quizzes	✗	✓	✓	✗	✗	✓
Report generation	✗	✓	✗	✗	✗	✓

System Overview Diagram



Methodology



Technologies used

- ▶ Keras (implement neural networks)
- ▶ React
- ▶ Firebase
- ▶ Google Colab (Training the model)
- ▶ Python(model)
- ▶ REST API
- ▶ Python flask (creating rest API)
- ▶ Open CV(image processing)
- ▶ Tensor Flow



REQUIREMENTS

FUNCTIONAL REQUIREMENTS

- ✓ Currency notes and coins Teaching Lessons
- ✓ Sentence Identifying Quizzes
- ✓ Show the correct answers
- ✓ Answer Evaluation & score

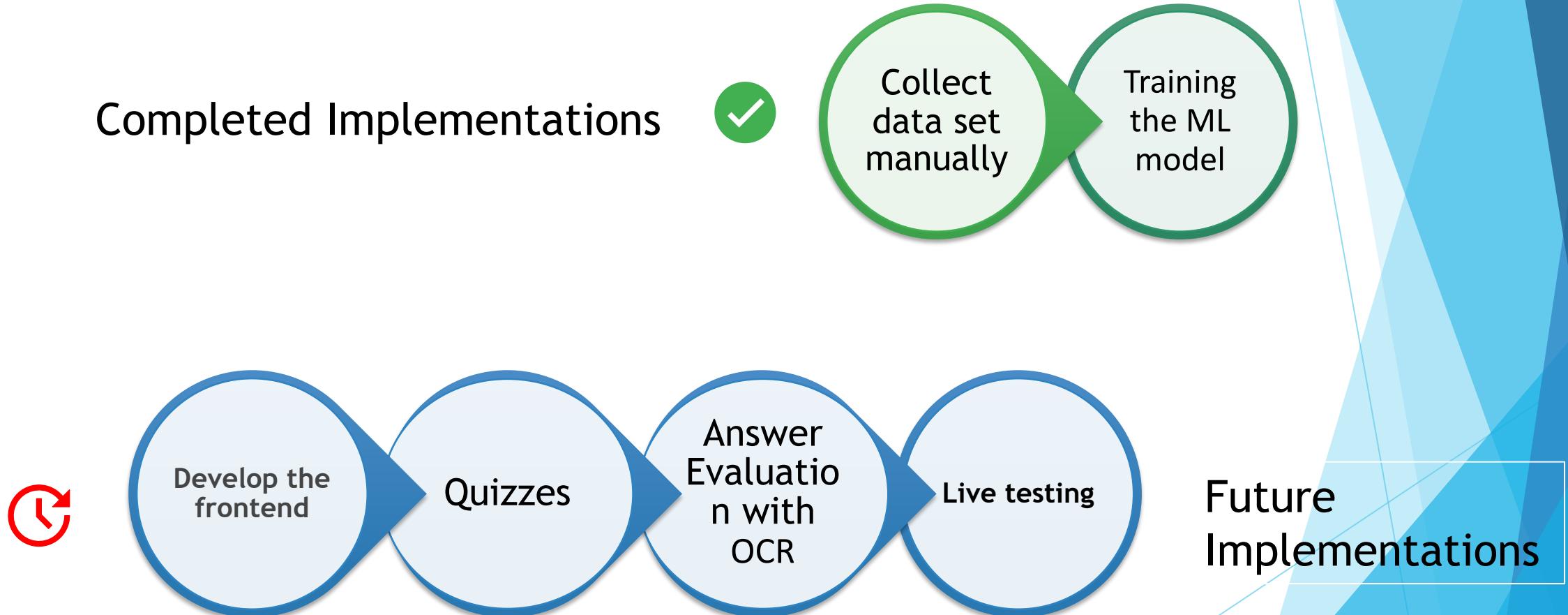
NON-FUNCTIONAL REQUIREMENTS

- ✓ Usability
- ✓ Reliability
- ✓ Portability
- ✓ Speed

USER REQUIREMENTS

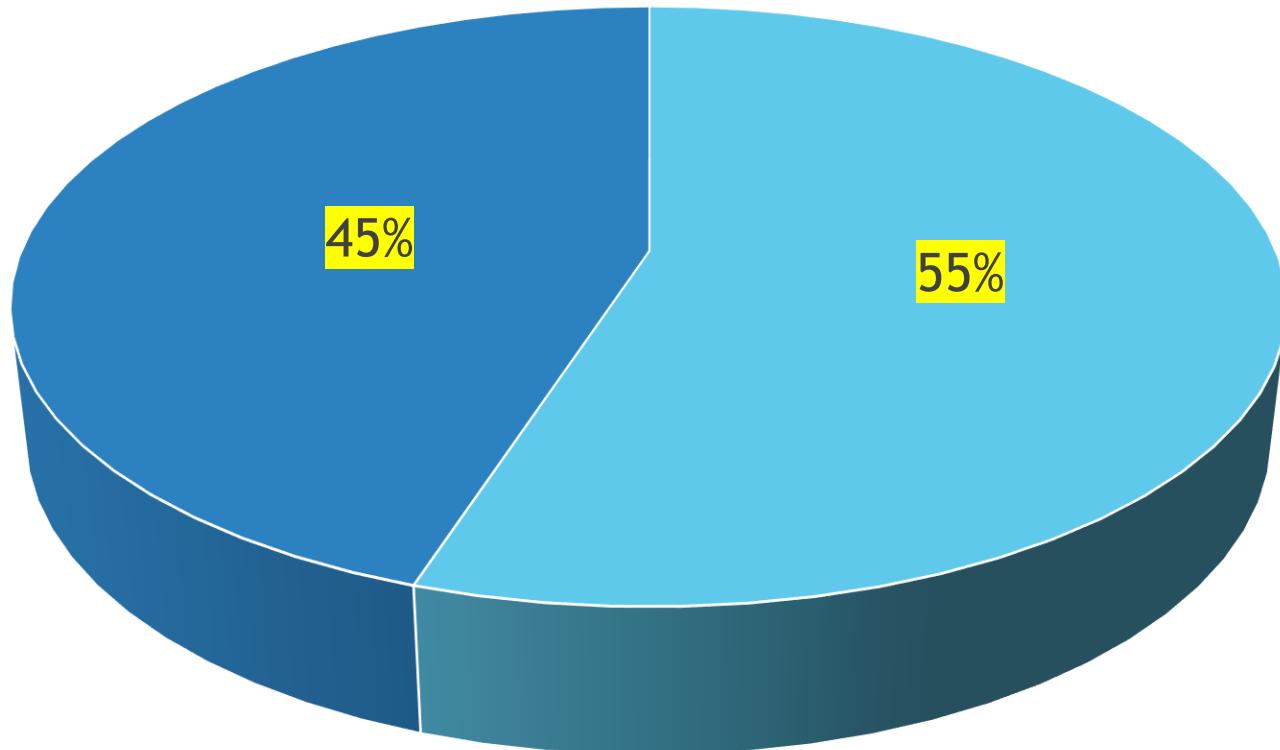
- ✓ Laptop
- ✓ Internet Connection
- ✓ Front Camera

COMPLETION AND FUTURE WORKS



Progress

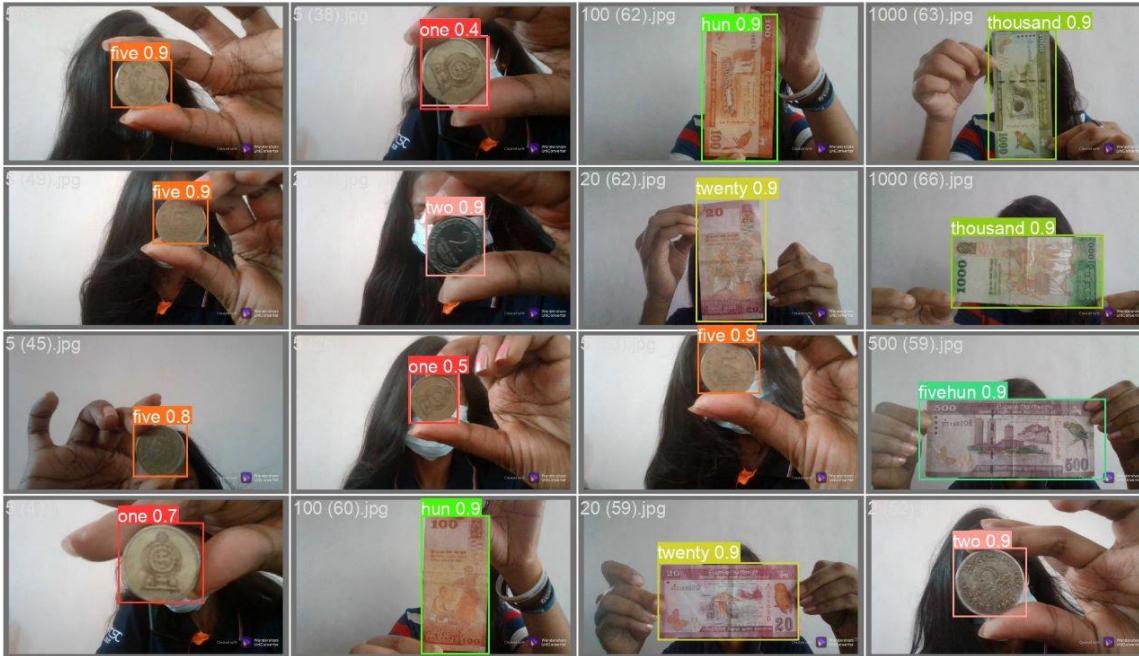
progress



■ pp1 ■ pp2

Dataset

Created a Dataset with,



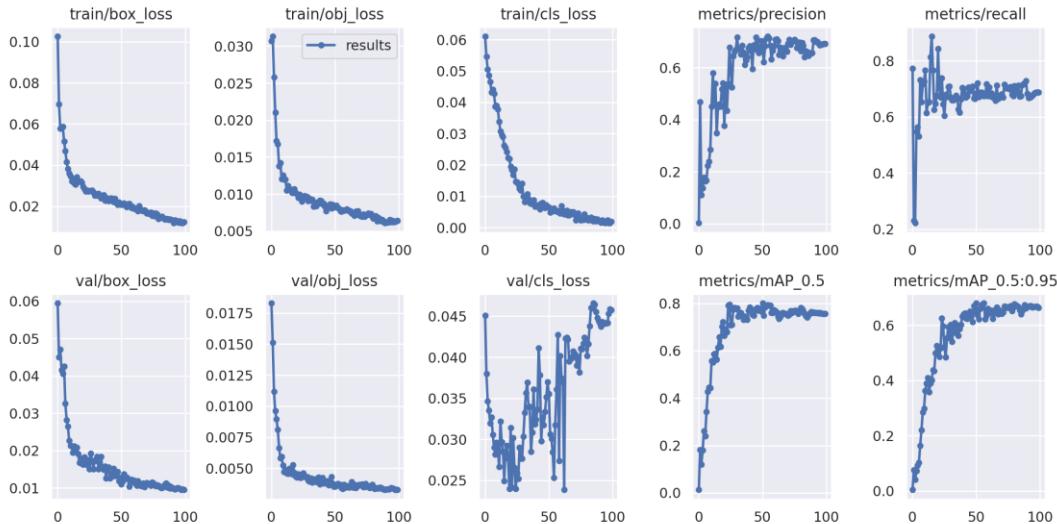
- Different backgrounds , currency notes and coin
- Use several coins and notes of the same type
- Took photos of coins and currency notes in various poses
- Use 200 images for one class



Training the model and Accuracy report



```
100 epochs completed in 0.182 hours.  
Optimizer stripped from runs/train/exp/weights/last.pt, 14.4MB  
Optimizer stripped from runs/train/exp/weights/best.pt, 14.4MB  
  
Validating runs/train/exp/weights/best.pt...  
Fusing layers...  
Model summary: 157 layers, 7031701 parameters, 0 gradients, 15.8 GFLOPs  
Class Images Instances P R mAP50 mAP50-95: 100% 4/4  
all 101 102 0.667 0.685 0.801 0.681  
one 101 15 0.815 0.884 0.942 0.698  
two 101 13 0.754 0.923 0.952 0.863  
five 101 12 0.879 0.608 0.834 0.621  
ten 101 8 0.858 1 0.995 0.886  
twenty 101 14 0.982 1 0.995 0.871  
hun 101 14 0.982 1 0.995 0.907  
fivehun 101 12 0 0 0.238 0.193  
thousand 101 14 0.0666 0.0619 0.458 0.408  
  
Results saved to runs/train/exp
```



- Model trained using Google Colab online tool

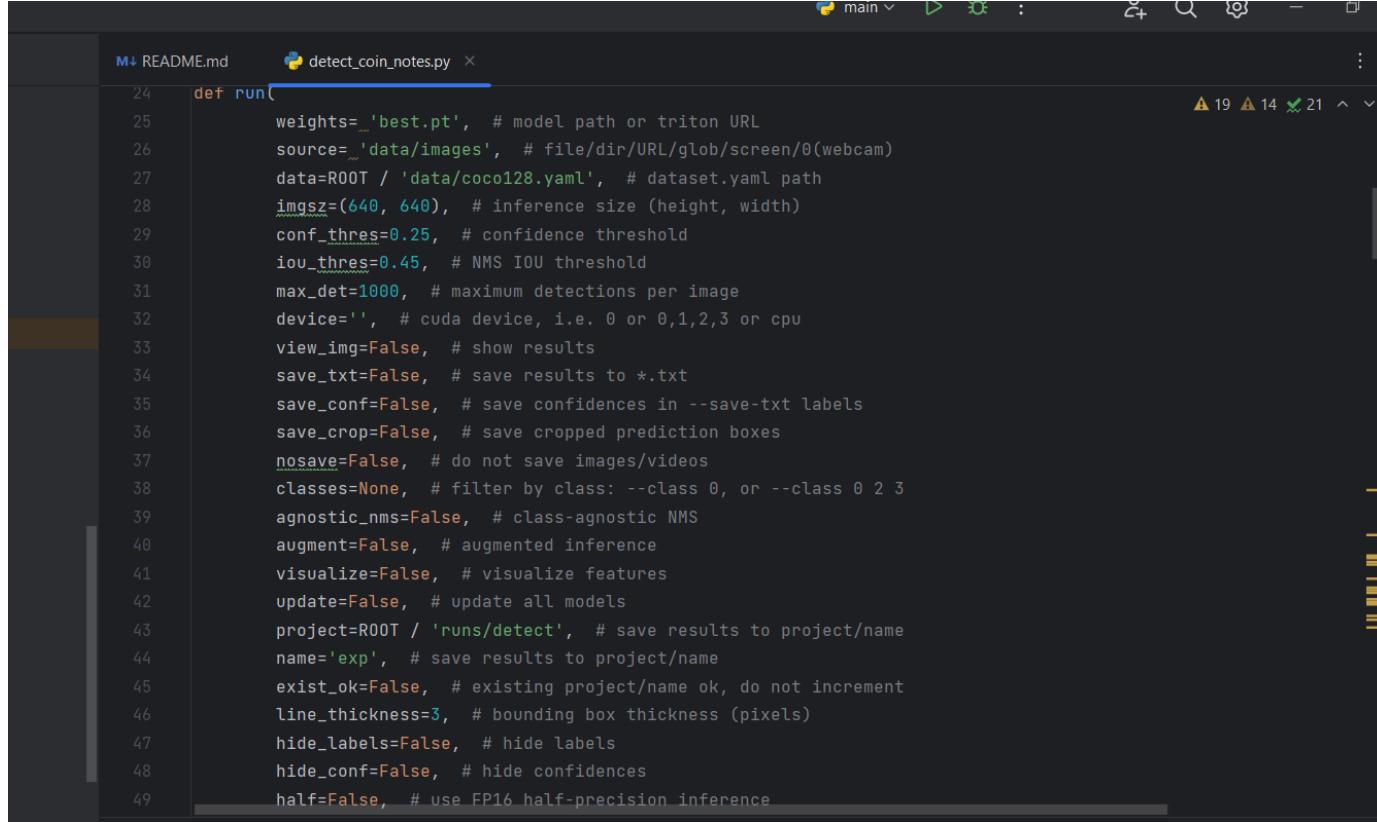
Accuracy: 0.9624

Loss: 0.113

Validation accuracy: 0.6533

Validation loss: 337.2601

Sign detection using camera input



A screenshot of a code editor showing a Python script named `detect_coin_notes.py`. The code defines a `run` function with various parameters for detecting currency notes and coins using a camera input. The code includes imports, model paths, dataset paths, inference sizes, confidence thresholds, NMS thresholds, maximum detections per image, device selection, result viewing, saving results, saving confidences, saving crops, and filtering by class.

```
def run(weights='best.pt', # model path or triton URL
       source='data/images', # file/dir/URL/glob/screen/0(webcam)
       data=ROOT / 'data/coco128.yaml', # dataset.yaml path
       imgsz=(640, 640), # inference size (height, width)
       conf_thres=0.25, # confidence threshold
       iou_thres=0.45, # NMS IOU threshold
       max_det=1000, # maximum detections per image
       device='', # cuda device, i.e. 0 or 0,1,2,3 or cpu
       view_img=False, # show results
       save_txt=False, # save results to *.txt
       save_conf=False, # save confidences in --save-txt labels
       save_crop=False, # save cropped prediction boxes
       nosave=False, # do not save images/videos
       classes=None, # filter by class: --class 0, or --class 0 2 3
       agnostic_nms=False, # class-agnostic NMS
       augment=False, # augmented inference
       visualize=False, # visualize features
       update=False, # update all models
       project=ROOT / 'runs/detect', # save results to project/name
       name='exp', # save results to project/name
       exist_ok=False, # existing project/name ok, do not increment
       line_thickness=3, # bounding box thickness (pixels)
       hide_labels=False, # hide labels
       hide_conf=False, # hide confidences
       half=False, # use FP16 half-precision inference
```

- Currency notes and coins
Detector class is implemented to identify the sign input through the camera

Ex.

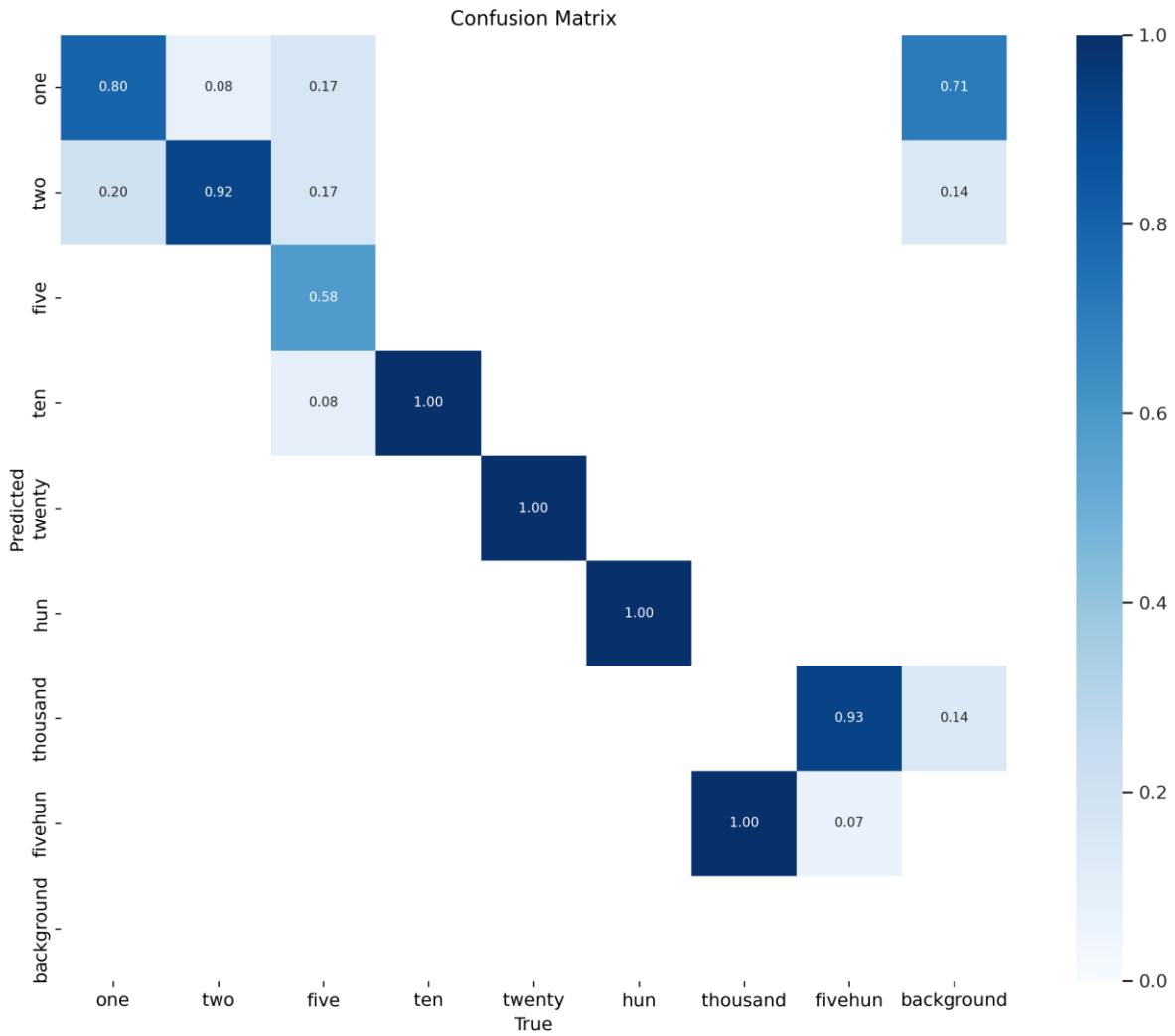


Detecting sign using camera

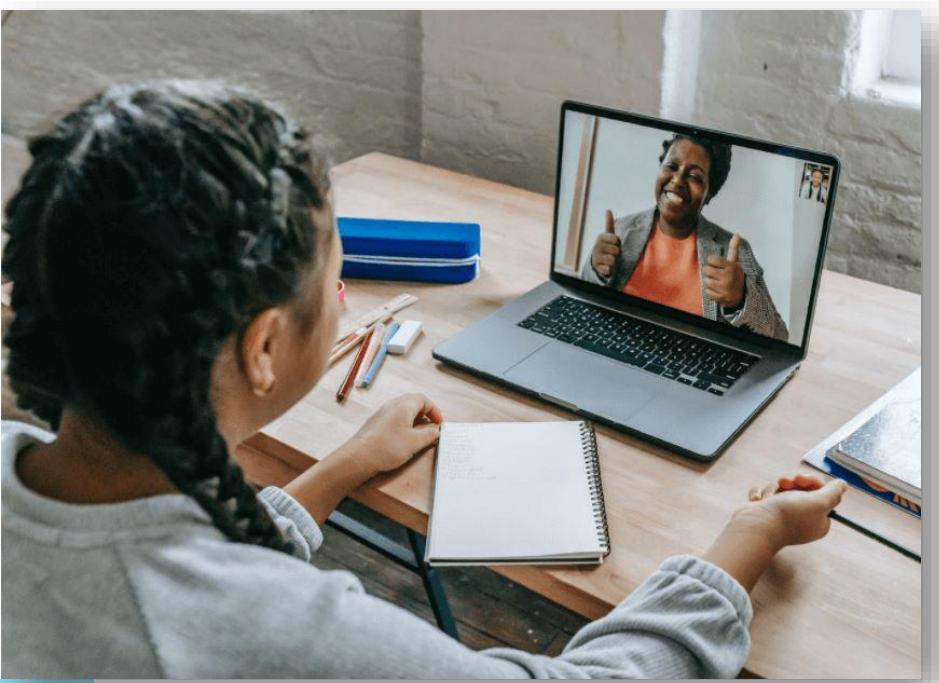
```
def parse_opt():
    parser = argparse.ArgumentParser()
    parser.add_argument('--weights', nargs='+', type=str, default=ROOT / 'models/best_coin.pt', help='model path or')
    parser.add_argument('--source', type=str, default=ROOT / '0', help='file/dir/URL/glob/screen/0(webcam)').#data/j
    parser.add_argument('--data', type=str, default=ROOT / 'data/coco128.yaml', help='(optional) dataset.yaml path')
    parser.add_argument('--imgsz', '--img', '--img-size', nargs='+', type=int, default=[640], help='inference size')
    parser.add_argument('--conf-thres', type=float, default=0.25, help='confidence threshold')
    parser.add_argument('--iou-thres', type=float, default=0.45, help='NMS IoU threshold')
    parser.add_argument('--max-det', type=int, default=1000, help='maximum detections per image')
    parser.add_argument('--device', default='', help='cuda device, i.e. 0 or 0,1,2,3 or cpu')
    parser.add_argument('--view-img', action='store_true', help='show results')
    parser.add_argument('--save-txt', action='store_true', help='save results to *.txt')
    parser.add_argument('--save-conf', action='store_true', help='save confidences in --save-txt labels')
    parser.add_argument('--save-crop', action='store_true', help='save cropped prediction boxes')
    parser.add_argument('--nosave', action='store_true', help='do not save images/videos')
    parser.add_argument('--classes', nargs='+', type=int, help='filter by class: --classes 0, or --classes 0 2 3')
    parser.add_argument('--agnostic-nms', action='store_true', help='class-agnostic NMS')
    parser.add_argument('--augment', action='store_true', help='augmented inference')
    parser.add_argument('--visualize', action='store_true', help='visualize features')
    parser.add_argument('--update', action='store_true', help='update all models')
    parser.add_argument('--project', default=ROOT / 'runs/detect', help='save results to project/name')
    parser.add_argument('--name', default='exp', help='save results to project/name')
    parser.add_argument('--exist-ok', action='store_true', help='existing project/name ok, do not increment')
    parser.add_argument('--line-thickness', default=3, type=int, help='bounding box thickness (pixels)')
```

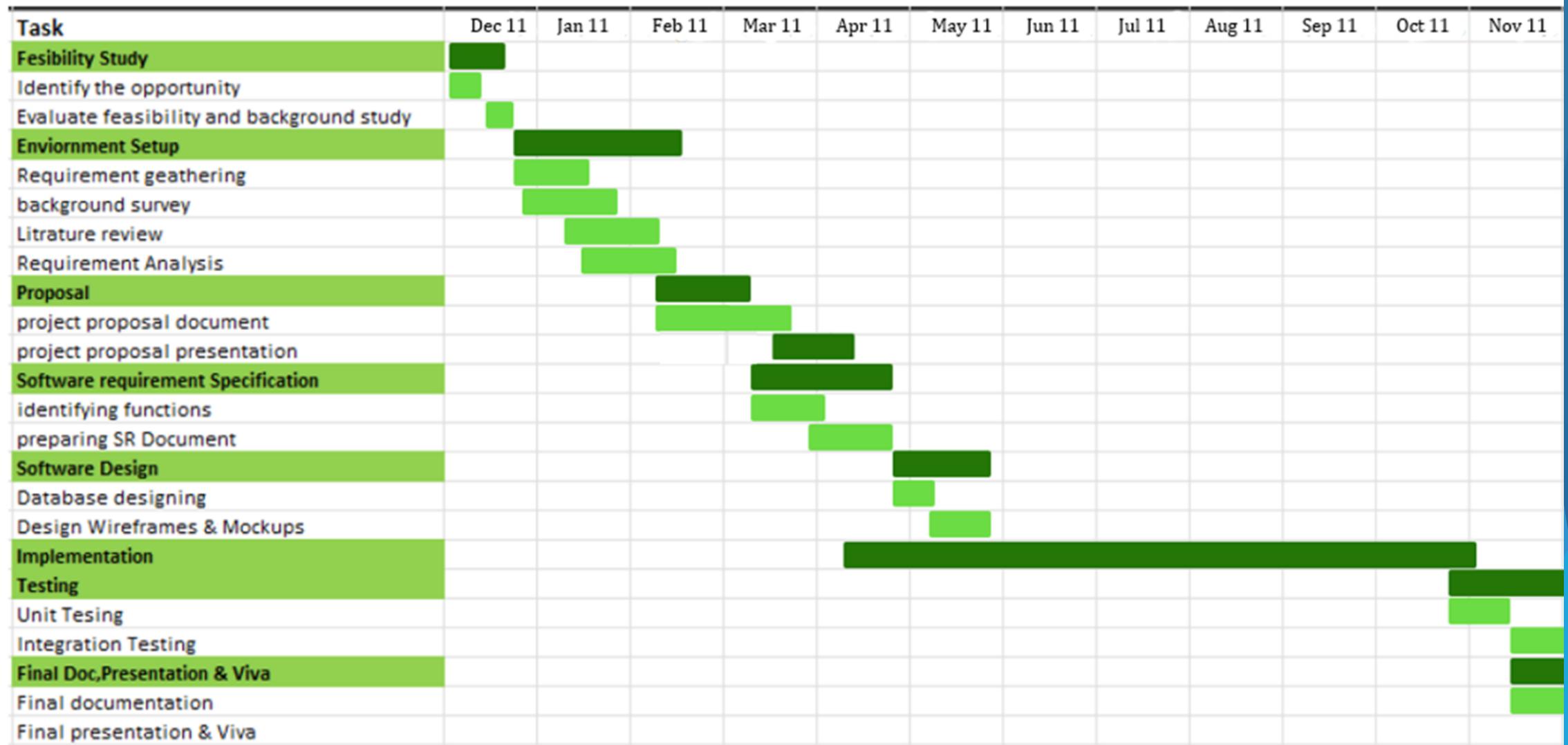
- Presence of the currency notes in the camera input is identified
- Image of the currency notes is used as an input for predicting the sign.
- Return the output to the web application

Accurecy check



Setup the webcam & Set up variables to keep track of the current sign and number of videos taken





THANK YOU

