

# EMBEDDED COIN RECOGNITION SYSTEM

2nd year of Computer Engineering Degree - End of Year Project

Presented by:

- Dhouha GAOUD
- Khalil OUALI

Proposed and supervised by:

- Prof. Kamel ECHAIEB

# AGENDA

1. Context
2. Design and Specifications
3. Chosen Technologies
4. Custom Model
5. Final Prototype
6. Conclusion and Prospects

# CONTEXT

## Objective:

**Helping the blind and the visually impaired recognize coins more easily and effectively**

- **Safer transactions**
- **Reduced dependence**
- **Increased privacy**



# CONTEXT

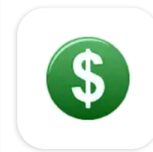
## Available solutions:

### The iBill Talking Bank Note Identifier



× Only for US paper bills

### Various mobile apps



**IDEAL U.S. Currency Identifier**

IDEAL Group, Inc. Android Development Team

× Only for US currency



**Coinoscope: Coin identifier**

Micron

Contains ads • In-app purchases

× **Not accessible for the visually impaired**

➤ **No viable solution for Tunisian blind people**

# CONTEXT

## Scope and limitations:

For the sake of prototyping:

- Non-functional requirements are not taken into consideration
- The system is only meant to recognize Tunisian coins
- Coins are recognized by group



Yellows



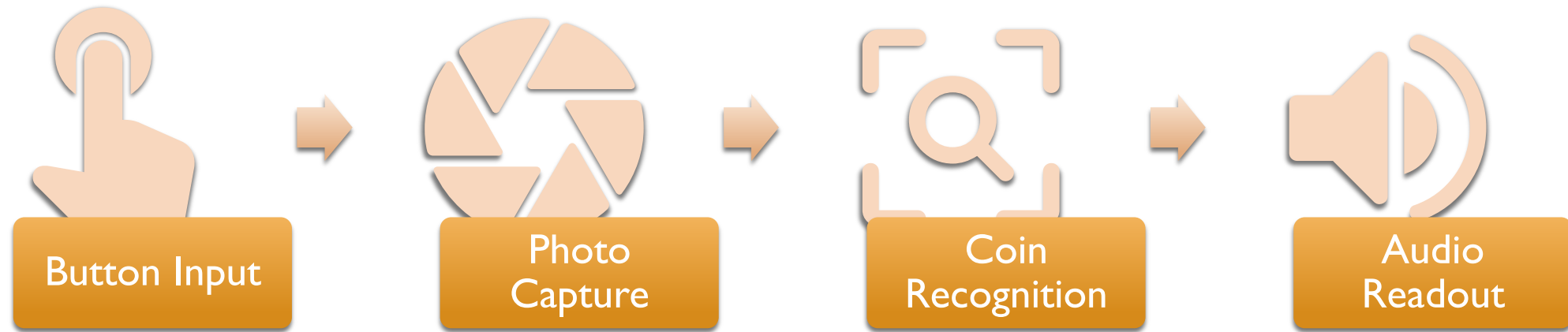
Silvers



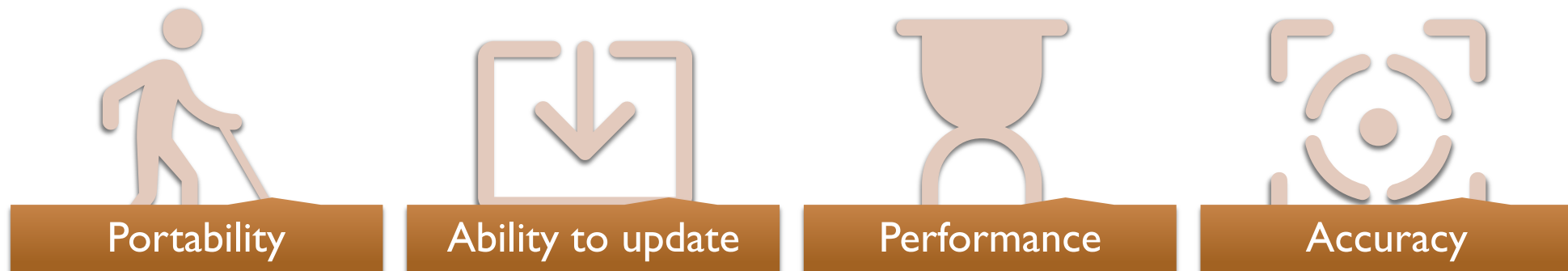
Fives

# DESIGN AND SPECIFICATIONS

## Functional requirements:

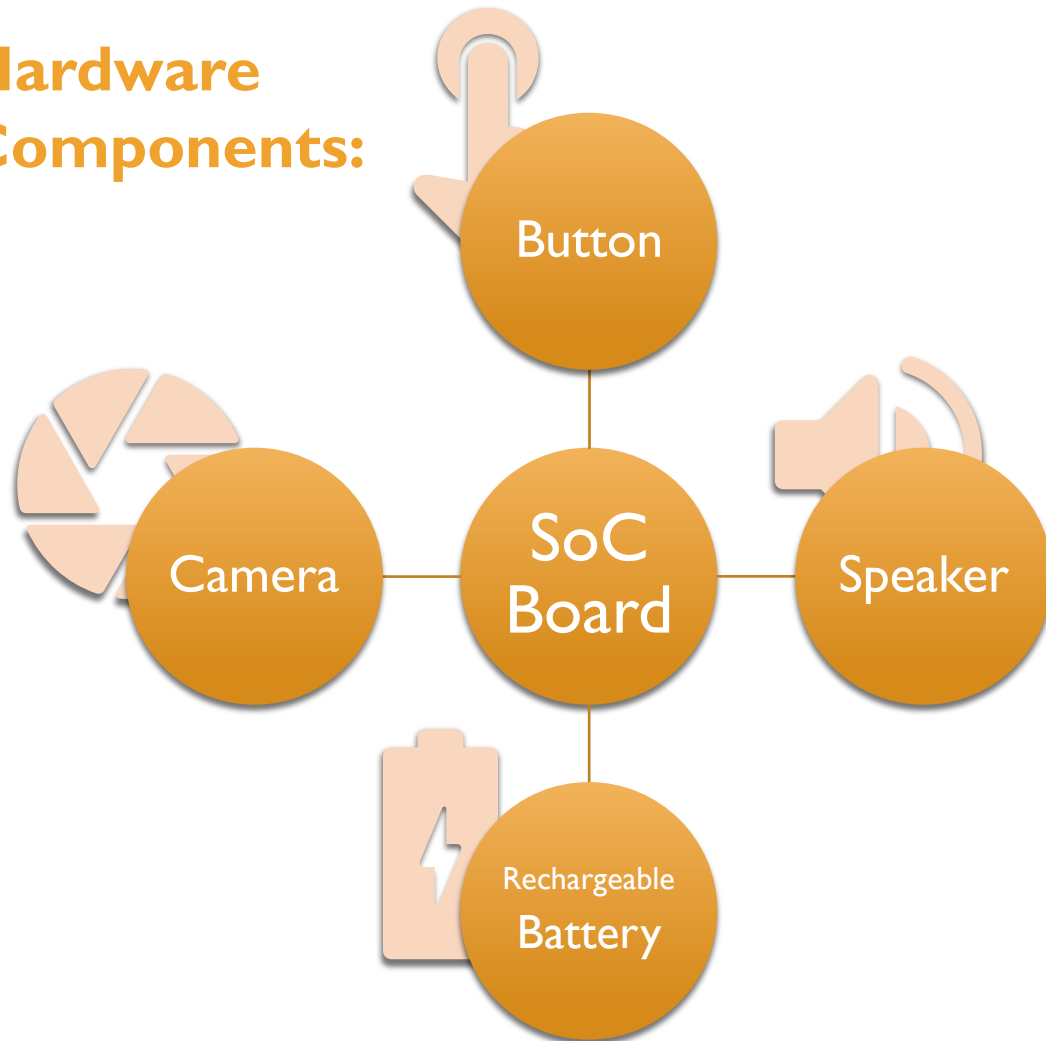


## Non-functional requirements:

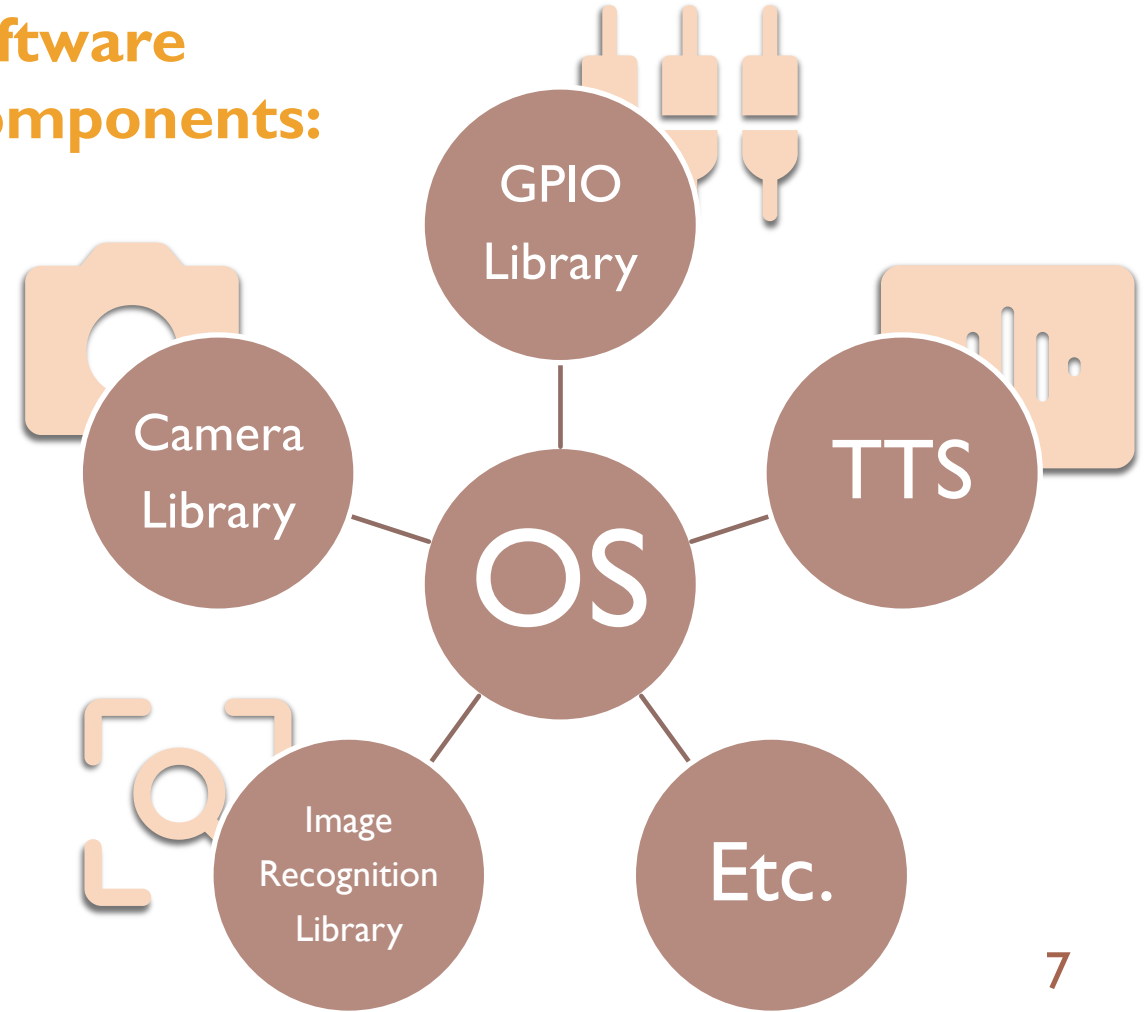


# DESIGN AND SPECIFICATIONS

## Hardware Components:



## Software Components:





---

# DESIGN AND SPECIFICATIONS

## Methodology:

1. Research potential technologies → Select hardware and software
2. Verify compatibility → Get a simple model working
3. Gather dataset and train a custom model
4. Implement a working prototype

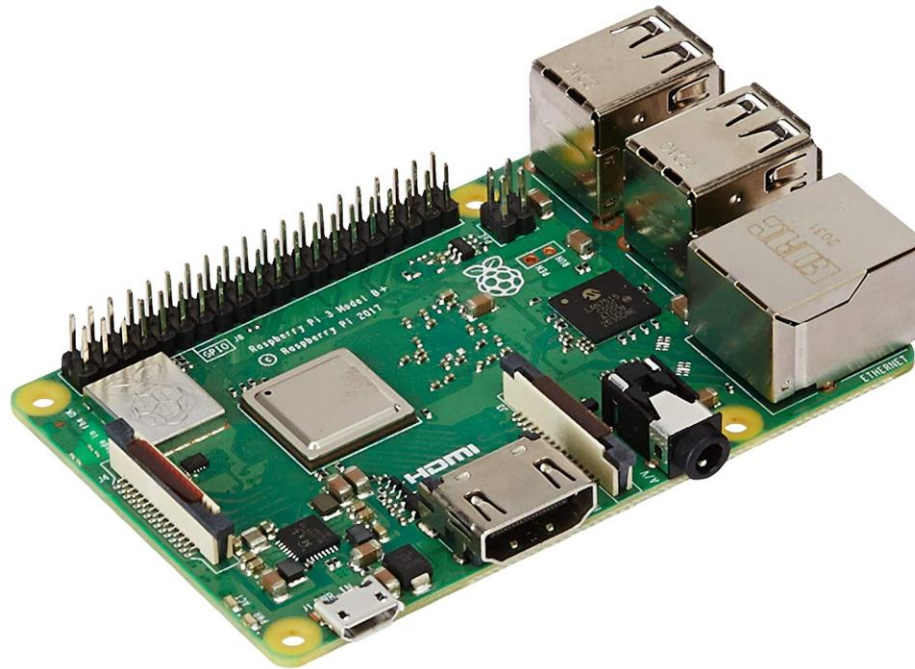


# CHOSEN TECHNOLOGIES

## Choice of Hardware:



ESP32-CAM



Raspberry Pi 3 Model B+



P5V04A SUNNY  
Raspberry Pi Camera



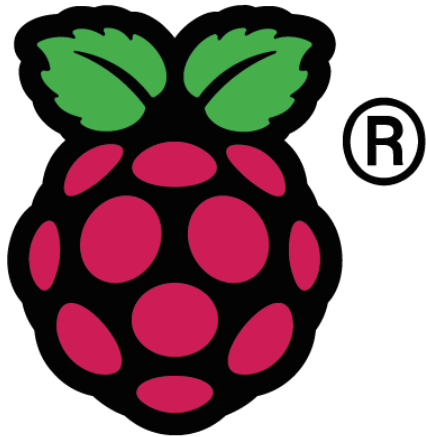
### Generic Components:

- Speaker
- Button
- Etc.

# CHOSEN TECHNOLOGIES

## Choice of Software

Main software:



**Raspbian Bullseye  
(64-bit) OS**



**Python 3**  
programming language

- ✓ Versatile
- ✓ Widely used



image recognition library  
based on  PyTorch

- ✓ Simple
- ✓ Unique solution



# CHOSEN TECHNOLOGIES

## Choice of Software Additional software:



for image manipulation



eSpeak NG

Text-to-Speech Synthesizer

- ✓ Light
- ✓ Works offline



Platform-specific packages:

- picamera2 (picamera)
- gpiozero



Miscellaneous packages:

- logging + datetime + time
- subprocess
- signal

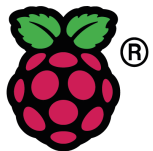


# CHOSEN TECHNOLOGIES

## Using Detecto on the Raspberry Pi 3B+

### Dependency issues:

- 1) 64-bit OS required
- 2) OpenCV (installed via **APT**)
- 3) PyTorch **v1.13.1** required



### Model considerations:

- Can not be quantized
- Must be based on *fasterrcnn\_mobilenet\_v3\_large\_fpn* due to hardware limitations

# CUSTOM MODEL

## Building the dataset

### Photo Capture:



Capture Setup

134 images, each containing 9-12 coins

- Camera fixed in place
- Both coin faces
- Varied lighting
- Different backgrounds



Samples of Captured Dataset Images

# CUSTOM MODEL

## Building the dataset

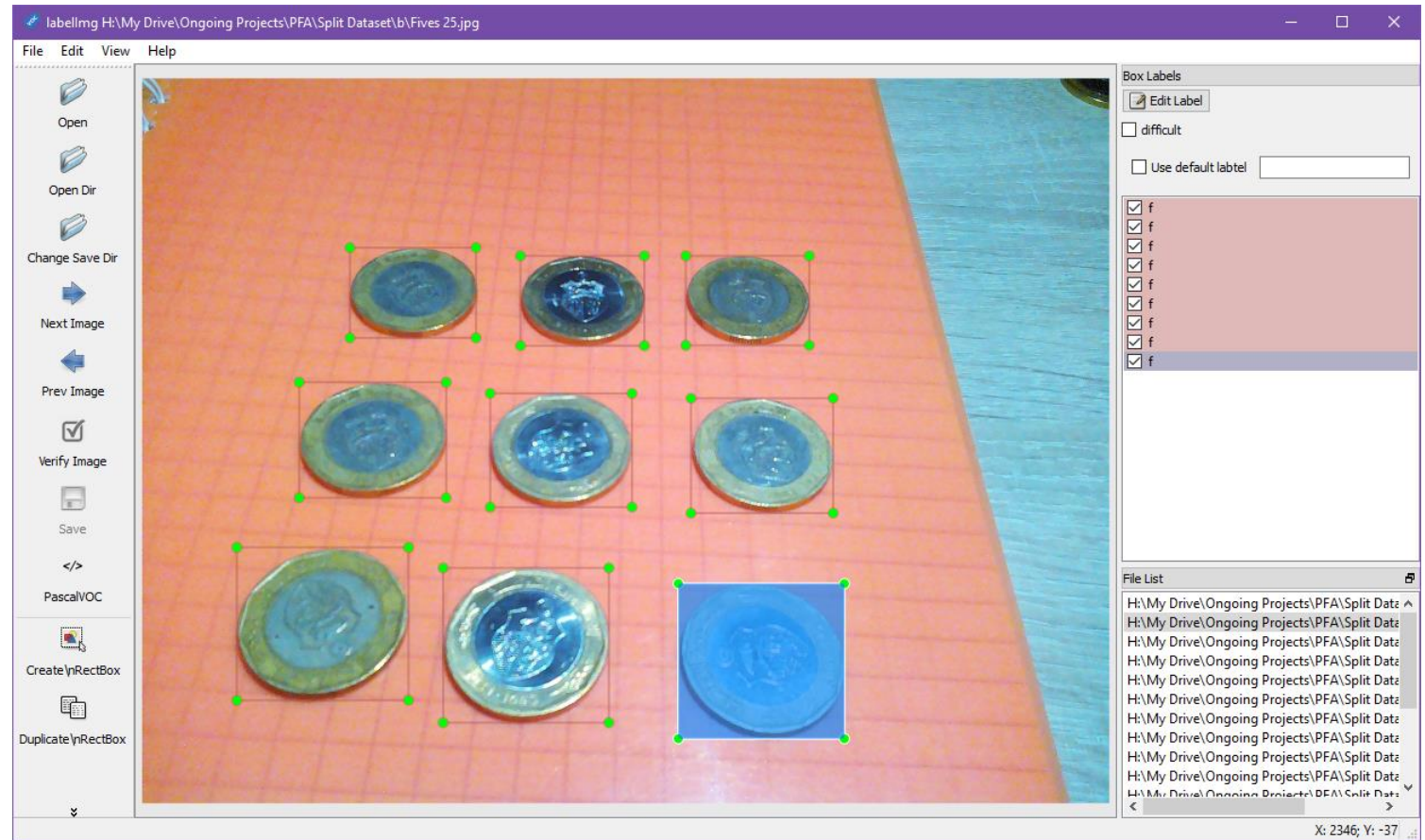
### Labeling:

#### Labeling Software: **Labellmg**

- ✓ Simple
- ✓ Widely used

#### 3 Classes:

- y (Yellow coins)
- s (Silver coins)
- f (Five dinar coins)



Screenshot of Labellmg



# CUSTOM MODEL

## Training Hardware:

### Option 1: Cloud



Google Colaboratory

- × Dataset upload times
- × Model download times
- × Slow training

### Option 2: Local



+ PyTorch

+ Detecto

On



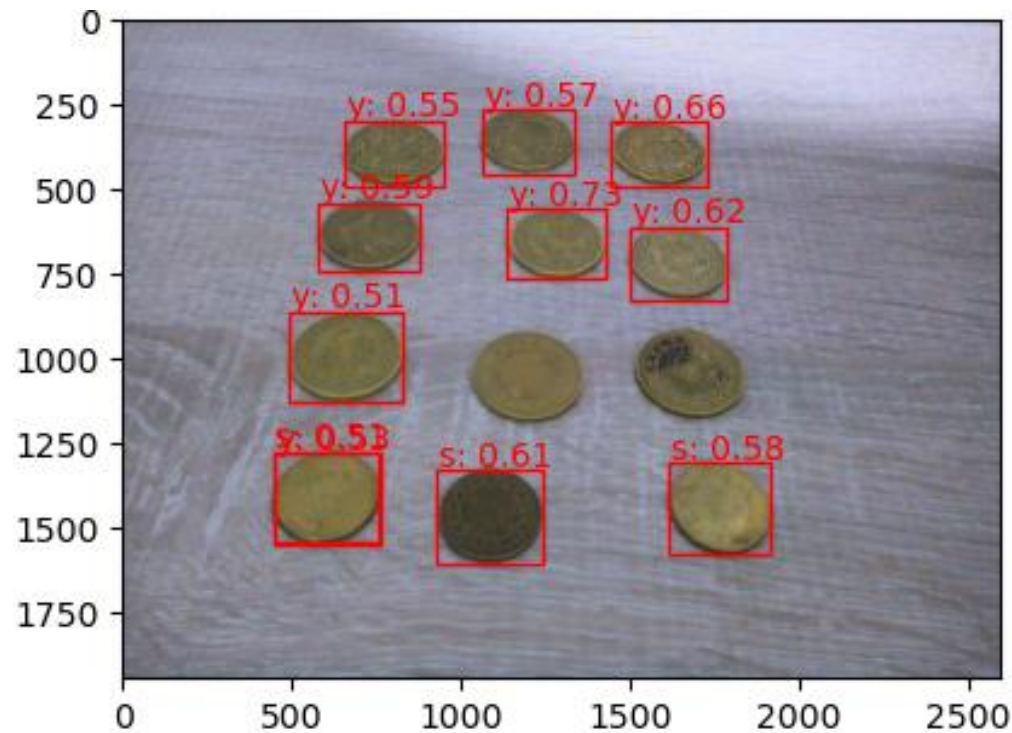
RTX3060 laptop GPU

➤ Overall faster and more convenient



# CUSTOM MODEL

## Training Results:



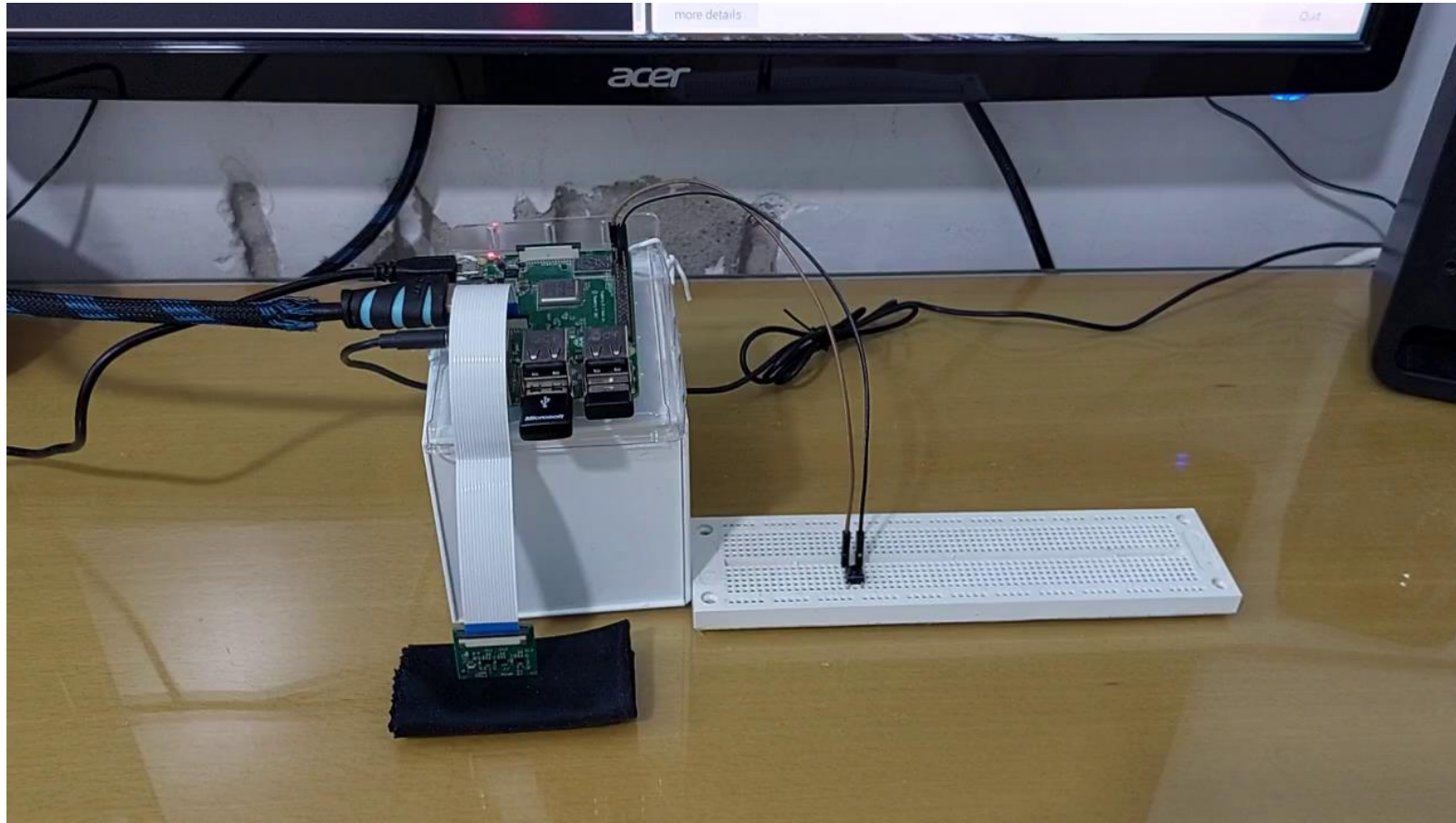
### Multiple models trained with different

- Batch size
  - Number of epochs
  - Dataset subsets
  - Augmentation functions
- Unsatisfactory results

### Likely factors:

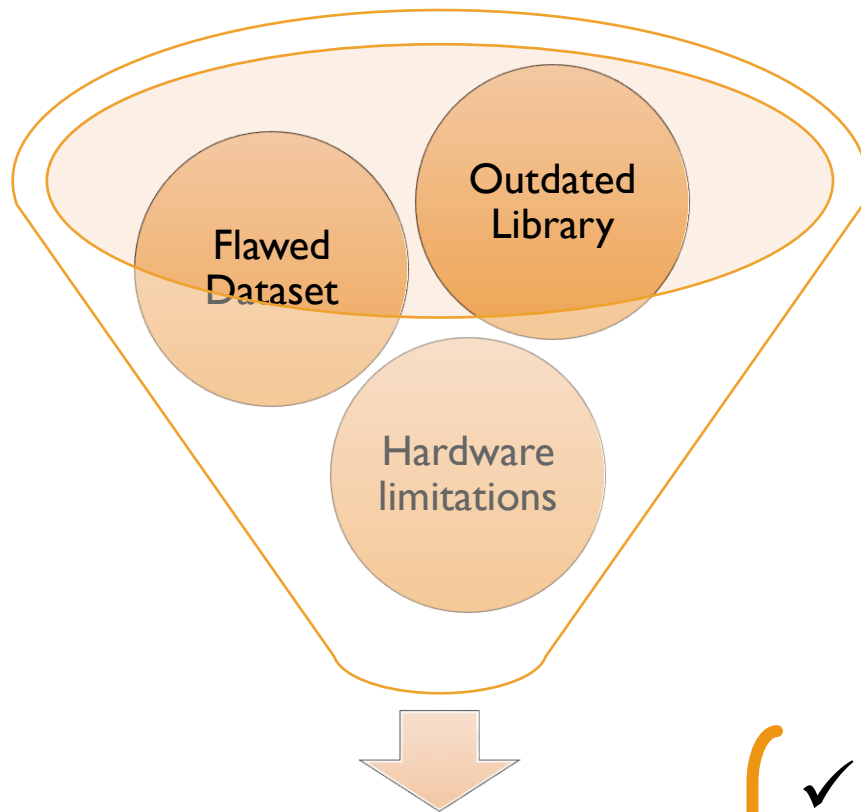
- × Limitations with Detecto
- × Limitations with the base model
- × Flawed dataset

# FINAL PROTOTYPE



Demonstration Video

# CONCLUSION AND PROSPECTS



Final Prototype {  
✓ Functional  
× Slow  
× Inaccurate



**Alternative:**



**TensorFlow Lite**

*"A mobile library for deploying models on mobile, microcontrollers and other edge devices."*



**Thank You**  
**for your attention.**