Indian Currency Recognition for Visually Impaired People

First-Level Project Presentation

Kevin S Monachan

Department of MCA

Guide: **Smt. Premy P Jacob**

Introduction

Introduction

This project is designed to assist visually impaired individuals in recognizing and managing Indian currency notes independently. It provides a **voice-guided system with audio feedback**, enabling users to identify currency and track amounts without needing visual cues.

Problem Statement

Problem Statement

A user-friendly, accessible solution for independent recognition and handling of Indian currency through non-visual means.

Project Objective

Project Objective

- Provide a reliable currency recognition service
- Enable real-time auditory feedback
- Maintain a digital virtual purse
- Offer an accessible, keyboard- and voice-driven interface
- Enhance financial independence and security

Goal: To develop an accessible system that empowers users to recognize and manage currency independently and securely.

Literature Survey (Part 1)

Paper Title	Methodology	Advantages	Disadvantages
A Robust System for Indian Paper Currency Recognition using Deep Learning	Fine-tuned VGG-16 CNN with transfer learning and image augmentation.	High accuracy, real-world robustness.	Computationally heavy; unsuitable for low-end devices.
Deep Learning Based Currency Recognition and Verification	Custom CNN trained on raw pixel inputs for denomination classification.	Good accuracy, fast inference with pre-trained layers.	Model size too large for real-time embedded applications.
SURF-Based Indian Currency Recognition	SURF feature extraction and keypoint matching with currency DB.	Low resource usage; fast detection.	Performance drops under varied lighting or distortion.

Literature Survey (Part 2)

Paper Title	Methodology	Advantages	Disadvantages
Fake Banknote Detection Using Machine Learning	Used statistical features from wavelet images; Random Forest gave best accuracy for fake note detection	High accuracy; portable and affordable hardware.	Requires special sensors; not ideal for large-scale classification.
A Survey on Paper Currency Recognition Systems	Comprehensive review of classification, detection, and feature extraction methods across studies.	Broad coverage of technologies; comparative insights.	No original model or implementation details provided.

Existing System Limitations

Current System Issues

- Relies on faded tactile marks
- Depends on subtle differences in size and texture
- Limits user independence and privacy

Proposed System

Proposed System

- Deep learning model based on ResNet architecture
- Currency recognition via webcam or image upload
- Voice command integration using Web Speech API
- Keyboard-driven interface requiring no mouse
- Continuous audio feedback for seamless interaction
- Virtual purse to track total currency amount

Tech Stack:

TensorFlow, ResNet, HTML, CSS, JavaScript, Web Speech API, Flask

Methodology

Data Collection & Preprocessing

- **Dataset:** Collected from Kaggle; images under varied lighting, angles, backgrounds.
- **Image Standardization:** Resized to 224×224 pixels, normalized, RGB preserved.
- **Label Encoding:** Folder-based class extraction and one-hot encoding.
- **Data Augmentation:** Rotation, flipping, brightness variation, and shifting (training set only).

Methodology

Model Architecture

- **Base Model:** Pre-trained ResNet50 (ImageNet).
- **Custom Layers:** Dense output layers tailored for Indian currency.
- **Loss Function:** Categorical Cross-Entropy:

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

- **Optimizer:** Adam optimizer used for adaptive learning.

Methodology

Two-Phase Training Strategy

- **Phase 1 – Feature Extraction:** Freeze base layers, train top layers (7 epochs, LR $= 10^{-4}$).
- **Phase 2 – Fine-Tuning:** Unfreeze last 30 layers of ResNet50, fine-tune (8 epochs, LR $= 10^{-5}$).

  Dropout Layers: Dropout applied (rates 0.5, 0.3, 0.2) to prevent overfitting.

Methodology

Callbacks & Regularization

- **Early Stopping:** Stops training if validation loss doesn't improve for 5 epochs.
- **ReduceLROnPlateau:** Lowers learning rate on plateaued validation loss.
- **Model Checkpoint:** Saves model with highest validation accuracy.

Methodology

Evaluation Metrics

- **Test Accuracy:** Overall correct predictions.
- **Top-3 Accuracy:** True label in top 3 predictions.
- **Classification Report:** Shows per-class precision, recall, and F1-score.
- **Confusion Matrix:** Displays confusion between similar denominations.

Model Architecture

- **Base Model:** ResNet50 (pre-trained on ImageNet) used for transfer learning.
- **Custom Head:** Fully connected layers added for Indian currency classification.
- **Loss Function:** Categorical Cross-Entropy:

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

- **Optimizer:** Adam optimizer for adaptive learning.

Two-Phase Training

- **Phase 1:** Freeze ResNet base; train only top layers for 7 epochs at 10^{-4} learning rate.
- **Phase 2:** Unfreeze last 30 layers of ResNet; fine-tune for 8 epochs at 10^{-5} learning rate.

Dropout: Added at 0.5, 0.3, and 0.2 in custom layers to reduce overfitting.

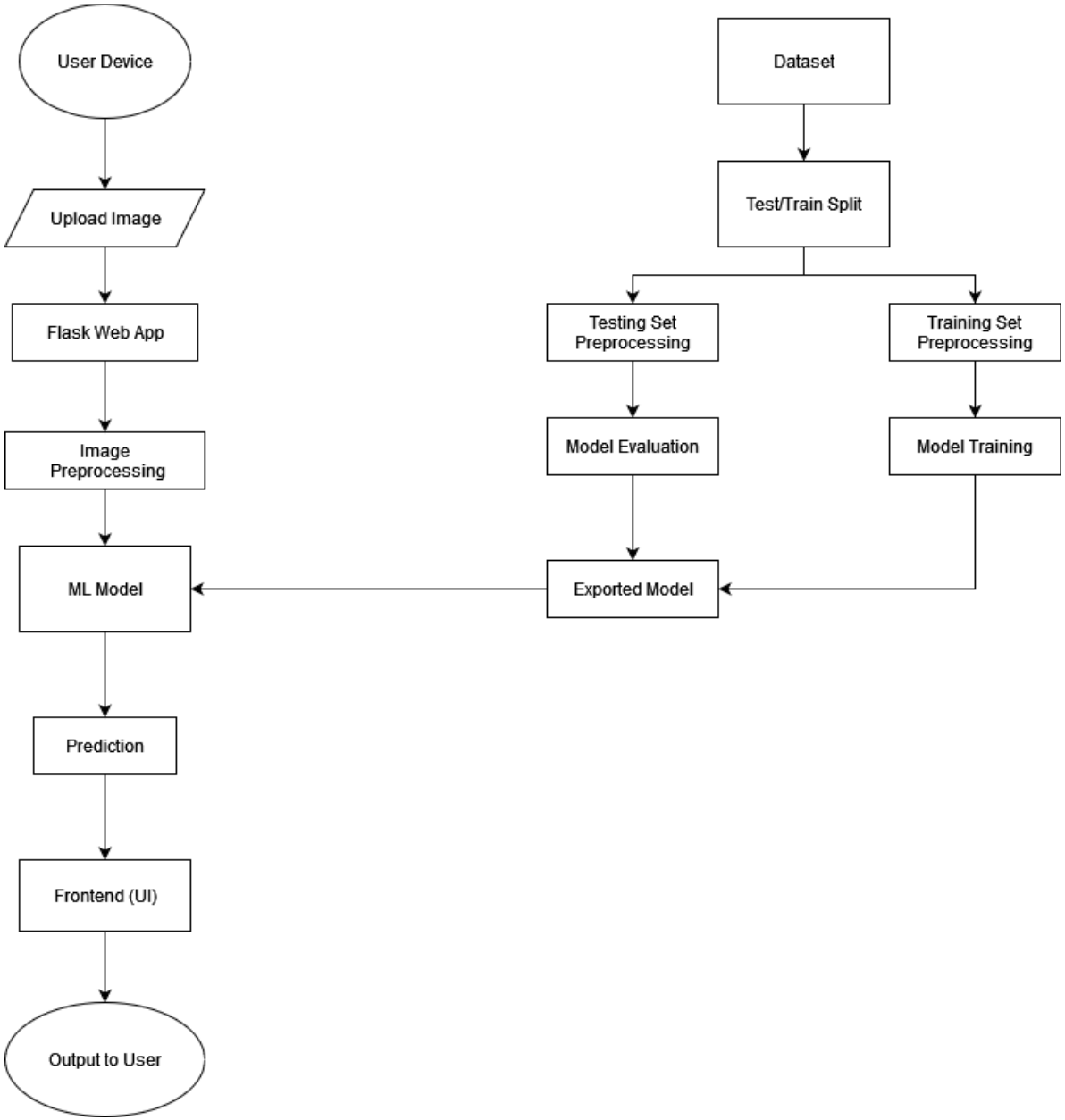
Callbacks & Regularization

- **Early Stopping:** Monitors validation loss, stops if no improvement in 5 epochs.
- **ReduceLROnPlateau:** Decreases learning rate on validation loss plateau.
- **Model Checkpoint:** Saves the best model based on validation accuracy.

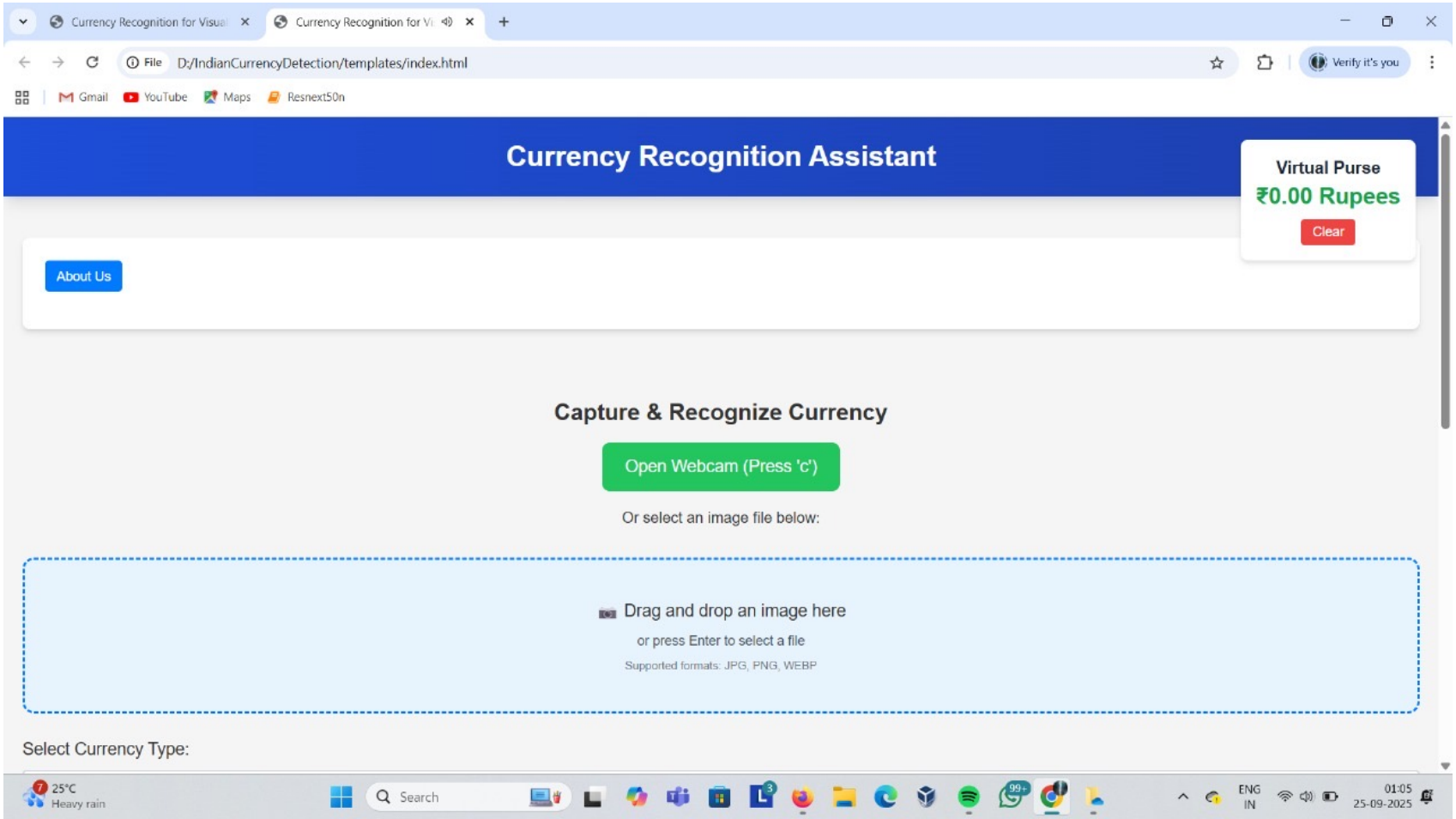
Evaluation Metrics

- **Test Accuracy:** Overall correct classifications on the test set.
- **Top-3 Accuracy:** Measures if correct label is in top 3 predictions.
- **Classification Report:** Includes precision, recall, and F1-score per class.
- **Confusion Matrix:** Shows misclassifications between denominations.

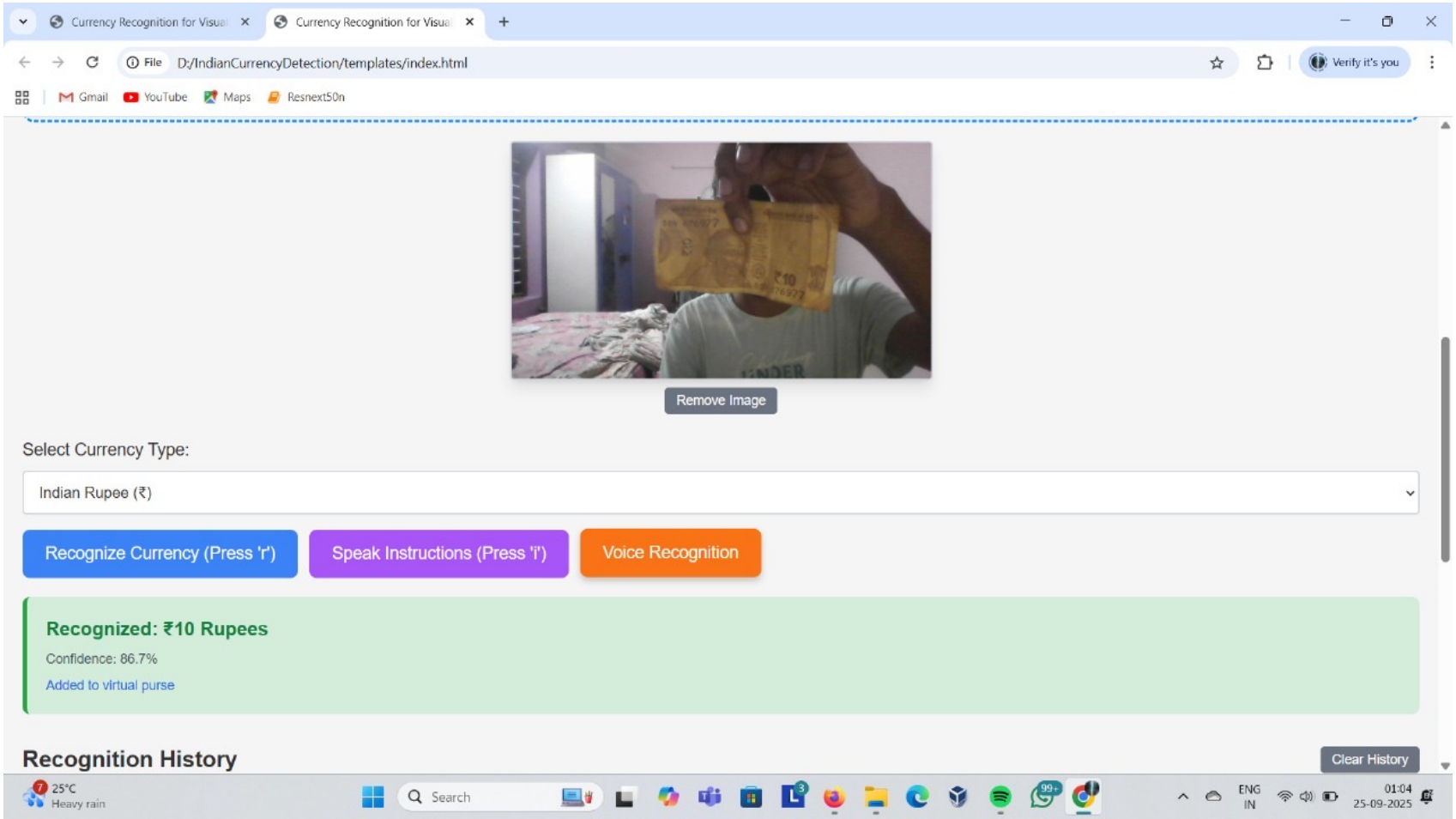
System Design / Architecture



Implementation - UI Screenshot 1



Implementation - UI Screenshot 2



System Configuration

System Configuration

Hardware Configuration

- **Operating System:** Windows
- **Processor:** AMD Ryzen 5 5600H
- **Memory:** 8GB RAM

Software Configuration

- **Language:** Python
- **Machine Learning Library:** TensorFlow
- **Model Architecture:** ResNet (CNN)
- **Front End:** HTML, CSS3, JavaScript, Web Speech API
- **Back End:** Python Flask

Conclusion

Conclusion

This project empowers visually impaired individuals to independently identify and manage currency, promoting financial autonomy and confidence. By prioritizing accessibility and inclusion, it supports a more equitable and dignified digital experience for all.

GitHub Repository

GitHub Link

github.com/Kevin-Monachan/CurrencyDetection

Thank You!