Handwritten Roman Numeral Recognition using MobileNetV2 and Web Deployment with Flask

Team Members:

Abdul Ahad 2023004

Ahmed Abubakar Riaz 2023902

Muhammad Hamza Khan 2023425

Muhammad Nabeegh 2023482

---

1. Introduction

The recognition of handwritten Roman numerals poses a challenging computer vision task due to the similarity of characters and variations in handwriting. In this project, we aimed to develop a deep learning-based classifier that accurately recognizes Roman numerals (I to X) from images and deploy it through a simple web interface.

---

2. Objectives

- To classify handwritten Roman numerals (I to X) using a CNN-based approach.
- To use transfer learning with MobileNetV2 for better performance on small datasets.
- To evaluate the model using standard metrics and a confusion matrix.
- To deploy the trained model as a Flask web application.

---

3. Dataset

- Images of handwritten Roman numerals (from I to X).
- Each class (i, ii, ..., x) is stored in separate directories.
- Dataset is split into training, validation, and test sets.

---

4. Model Architecture and Training

Approach

We used a transfer learning approach with MobileNetV2 as the base model. The base model was pre-trained on ImageNet and its weights were frozen initially. We added custom dense layers on top:

```
x = base_model.output

x = GlobalAveragePooling2D()(x)

x = Dense(64, activation='relu')(x)

predictions = Dense(len(class_labels), activation='softmax')(x)

model = Model(inputs=base_model.input, outputs=predictions)
```

Compilation

```
model.compile(optimizer=Adam(learning_rate=0.0001),
        loss='categorical_crossentropy',
        metrics=['accuracy'])
```

Training

- Model trained for 15 epochs
- Training Accuracy (final epoch): ~90.7%
- Validation Accuracy: ~84.2%
- Training Loss: 0.2779
- Validation Loss: 0.4351

Saving the Model

```
model.save("model.h5")
```

---

5. Evaluation Metrics

Final Test Accuracy:

accuracy: 0.8793

loss: 0.3360

Classification Report:

- Overall Accuracy: 84%
- Average Precision: 0.85
- Average Recall: 0.83
- Average F1-Score: 0.83

Confusion Matrix Insights:

- Model performed well on distinct numerals like I, V, X.
- Confusions were common in visually similar numerals (iv, vi, vii, ix).

---

6. Model Weaknesses and Error Analysis

We analyzed the confusion matrix and found key weaknesses:

- **VI** was misclassified as IV, IX, VII due to shared characters.
- **VII** was often confused with III.
- **IV** and **IX** were highly interchangeable in prediction due to structural similarity.

Recommendations for Improvement:

- Increase dataset diversity with more handwriting styles.
- Use data augmentation.
- Fine-tune more layers of the base model.
- Experiment with focal loss or class weights.

---

7. Web Deployment with Flask

We developed a Flask web app to allow users to upload a Roman numeral image and get a prediction.

Flask Code Highlights (app.py)

- Loads model with load_model("model.h5")
- Handles file uploads
- Preprocesses image using preprocess_input()
- Predicts and displays result

HTML Interface

- Simple upload form
- Displays predicted numeral

---

8. Conclusion

This project successfully demonstrated how transfer learning with MobileNetV2 can be used for a relatively small but complex task like handwritten Roman numeral recognition. Despite challenges due to the structural similarity of Roman numerals, we achieved an accuracy of 84% and deployed the model in a user-friendly Flask web app.
