Tinkerers' Lab ML Coordinator Selection Tasks

April 28th, 2024

Tinkerers' Lab, IIT Hyderabad
Between A, B, G, H Hostel Block, IIT Hyderabad
IITH Main Road, Near NH-65, Sangareddy, Telangana 502285

Tinkerers' Lab
IIT Hyderabad

Contents

1 Introduction

You are provided with a dataset containing images of coins and a corresponding CSV file that maps each image to its country of origin and denomination. Your task is to develop a deep learning model that can classify coins based on visual appearance alone. This involves training a model that can handle variations in lighting, wear, background noise, and orientation.

2 Objective

The expected output for each image is a predicted label that correctly identifies the coin's country and denomination (e.g., "India 10 Rupees", "USA Quarter"). The model will be trained on the provided dataset and evaluated based on accuracy and class-wise performance metrics.

Dataset Link:
Link

3 Key Requirements

3.1 Data Splitting

- Split the dataset into training and validation sets.
- Use the training set for model learning and the validation set for hyperparameter tuning.
- Use the test set for final evaluation.

3.2 Model Selection

- Use Convolutional Neural Networks (CNNs) for robust feature extraction from images.
- Start with pretrained models such as ResNet or others.
- Explore fine-tuning versus using frozen CNN backbones.

3.3 Hyperparameter Tuning

Experiment with:

- Learning rate
- Batch size
- Number of fully connected layers

- Number of hidden units and dropout rates

3.4 Preventing Overfitting

Apply techniques such as:

- Data augmentation (e.g., rotation, brightness shift)
- Dropout layers
- Early stopping
- L2 regularization

3.5 Loss Function and Optimization

- Use Cross-Entropy Loss for multi-class classification tasks.
- Train the model using optimizers like Adam or SGD with momentum.

3.6 Evaluation Metrics

Evaluate the model on the test set using:

- Accuracy
- Precision, Recall, and F1-Score
- Confusion Matrix for class-wise insight

3.7 Documentation

Document the following:

- Dataset exploration and preparation steps
- Model architecture and rationale
- Training configuration and decisions
- Augmentation techniques and hyperparameter trials

3.8 Testing and Final Evaluation

- Test your trained model on the held-out test set.
- Analyze and report:
  - Misclassified examples
  - Accuracy per coin type
  - Performance bottlenecks and how they were addressed

4 Hints

4.1 CNN Feature Extraction

Use ImageNet-pretrained CNNs like ResNet18 or EfficientNetB0 to extract discriminative features from coin images.

4.2 Data Augmentation

Implement real-world augmentations:

- Random rotations
- Flipping and scaling
- Lighting variation
- Random occlusion

4.3 Calibration and Label Mapping

- Ensure consistent mapping of CSV labels to class indices.
- Combine fields (e.g., country + denomination) to create unified class names.

4.4 Monitor Training Progress

Track:

- Training/validation loss and accuracy
- Confusion matrix updates per epoch
- Misclassified samples after each training phase

5 Deliverables

5.1 Google Colab Notebook or Code File

A complete notebook that includes:

- Data preprocessing
- Model building and training
- Evaluation on the test set
- Visualization of metrics and predictions

5.2 Summary Report

Include:

- Final accuracy and class-wise metrics
- Challenges faced during training and how they were overcome
- Design choices and future improvements

5.3 Code and Results

Make sure the notebook:

- Contains all code needed to reproduce results
- Includes charts (loss/accuracy curves) and confusion matrix
- Saves trained model weights (if required)

5.4 Reflections

Discuss:

- What worked well vs what didn't
- Impact of augmentation and architecture choices
- What you'd improve with more time/data

All of the above tasks and concepts should be documented and explored directly in your Colab notebook.

Important Note: Please submit your progress even if you are unable to complete the task. It will be taken into consideration. Remember that the use of AI tools can be easily detected. Strive to work independently and refer to documentation when necessary.

Good luck!