COMP3314 Machine learning

Assignment 3

2025 HKU COMP3314 Image Classification Challenge

Name: Xiao Yuxuan 3035974389

Cheung Ching Pan 3036101721

Cheung Man Hei Kay 3036106109

1. Environment Requirements:  
   - Linux with NVIDIA GPU  
   - CUDA 12.0+ (recommended: 12.5+)  
   - Python 3.10-3.12  
   - RAPIDS Libraries *(cudf, cuml)*  
   - OpenCV, NumPy, Pandas, scikit-image, scikit-learn
2. Dataset analysis:
   1. Dataset Overview

The dataset consists of labeled image files for training and unlabeled files for testing:

* *train.csv* contains image names (*im\_name*) and their corresponding labels.
* *test.csv* contains image names for which predictions are required.
* Image files are located in *train\_ims*/ and *test\_ims*/.
  1. Data Augmentation

To improve the robustness of the classifier, the dataset was augmented using the following transformations:

* Horizontal flip
* Vertical flip
* 180° rotation

Each training image was expanded to 4 images, effectively quadrupling the dataset size.

* 1. Category Statistics

After loading the dataset, we found:

* Total training images (after augmentation): *4 × original\_count*
* Number of unique classes: *num\_classes = {len(set(labels))}*
  1. Visualization

One example from each class was visualized.

1. Classifier Exploration:

2.1 Feature Extraction: Custom Preprocessor

A custom *`Preprocessor`* was implemented using:

- Histogram of Oriented Gradients (HOG) features from each RGB channel

- Flattened resized image (16×16) combined with HOG features

- Combined into one feature vector

This hybrid approach captures both color texture and edge information

2.2 Classifier 1: Support Vector Machine (SVM)

* Library: *cuml.svm.SVC* (GPU-accelerated)
* Kernel: RBF
* Parameters: *C=30, gamma='scale', random\_state=42*
* Pipeline: Preprocessing → StandardScaler → SVC
* Training Time: *~ {end\_time - start\_time:.2f}* seconds
* Performance: Due to lack of validation split in final training, evaluation is based on visual inspection and submission results.

2.3 Classifier 2 : Random Forest Classifier

* Library: *`cuml.ensemble.RandomForestClassifier`\*
* Settings: *`n\_estimators=100`, `max\_depth=25`*
* Pipeline: Preprocessor → StandardScaler → RandomForest
* Evaluation: Performed using a validation split (80% train / 20% val)
* Validation Accuracy: *`~{rf\_accuracy:.4f}`*
* Observation: Random Forest showed strong performance with shorter training time and less sensitivity to feature scaling compared to SVM.

1. Final Solution Description:

3.1 Overview

The final solution uses a GPU-accelerated pipeline:



* 1. Key Components

- Preprocessing: Custom transformer using HOG from *scikit-image* and image flattening.

- Scaling: *cuml.preprocessing.StandardScaler()* ensures features are normalized.

- Classification: *cuml.svm.SVC()* provides high performance on GPU-enabled environments.

- Augmentation: Simple flipping and rotating significantly increased training diversity.

* 1. Inference

Test images were processed using the same pipline. Predictions were saved to *submission.csv* with the following format:

A screenshot of a computer

AI-generated content may be incorrect.

1. Notes and Recommendations:
   * + CUDA Compatibility: Ensured by using CUDA 12.0+ and RAPIDS cuDF/cuML 25.4.
     + Hardware Acceleration: The use of GPU (required by RAPIDS) significantly reduced training time.
     + No Neural Networks Used: Fully compliant with assignment rules.
     + No External Datasets or Pre-trained Models Used.