Image Classification Project Proposal

Ahmad Hasan (40294488) Waqas Mansoor (40220269) Shibin Koshy (40295019)

GitHub Repository : https://github.com/waqasmansoor/Image-Classification

Abstract **-** Our team, Devine AI, aims to classify images into five distinct venue categories using both traditional and machine learning approaches. We will utilize a dataset consisting of more than 3500 color images representing five classes (Restaurant, Library, Lakeside, Auditorium, and Golf course). The goal of our project is to compare the performance of supervised and semi-supervised decision trees alongside a convolutional neural network (CNN) in classifying the venues. By implementing various image preprocessing techniques and hyperparameter tuning, we will evaluate each model using metrics such as accuracy, precision, recall, F1-score, and confusion matrix to determine the most effective approach for venue classification.

Keywords - Convolutional Neural Network ,Semi-Supervised Tree ,PyTorch ,F1-score ,Decision trees

I. Dataset

The dataset consists of 3926 images in total, and they are categorized into 5 distinct classes. The [1]**Library** class contains 749 images, the [2] **Lake side** class contains 732 images, [3] the **Auditorium** class contains 811 images, [4] the **Golf course** class contains 787 images and [5] , [3] the **Restaurant** class contains 847 images.

Figure 1 shows a bar graph that represents the distribution of images across the different classes in the dataset. The bar graph displays the total number of images without splitting into Train, Test and Validation. We selected the image resolution 256 x 256 for the training and prediction.

II. Possible Methodologies

To classify venues, we will implement and compare supervised decision trees, semi-supervised decision trees, and CNN. Preprocessing steps include resizing images to 256x256 pixels. The supervised decision tree will use labeled data, while the semi-supervised tree will iteratively incorporate high-confidence predictions from unlabeled data. The CNN, implemented in PyTorch, will be optimized for hyperparameters like layer count and learning rates. The predictions would be made on unseen dataset (test) and we are expecting to achieve more than 80 % accuracy overall with CNN. We will do comparison by plotting Confusion Matrix, ROC Curve and F1-score.

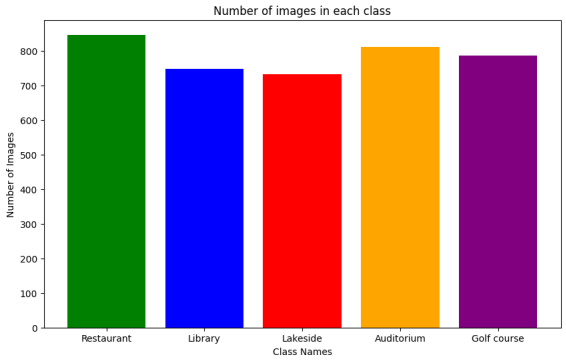

Figure 1 – Number of Images in each class

III. Data Pipeline

We will use color images as the input for our model. Since the images are taken from different sources, we will pass all the images through a resize method. The images would be shuffled and splitted into Train, Test and Validation dataset by ratio 70:20:10. We will use a data loader to load the images in batches. Our best guess is that the batch size would be 8 for training and for testing we will take 12 images per batch. We [3]decided to use a validation dataset during training to measure the accuracy during training and to do debugging. To achieve higher accuracy, we are planning to use the following techniques:

- Batch Normalization
- Dropout
- Data Augmentation
- Increasing the Number of Layers and epoch.

IV. Reference

[1] https://github.com/emanhamed/Houses-dataset. [Online].

[2] https://images.cv/dataset/lakeside-image-classification-dataset. [Online].

[3] https://images.cv/dataset/auditorium-image-classification-dataset. [Online].

[4] https://images.cv/dataset/library-image-classification-dataset. [Online].

[5] https://www.kaggle.com/datasets/nickj26/places2-mit-dataset. [Online].

[6] https://images.cv/dataset/restaurant-image-classification-dataset. [Online].