Image Recognition Model for Venue Classification Using Decision Trees

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

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

1. Introduction and Problem Statement

The aim of this project is to develop an image recognition model that classifies images into five venue categories: Restaurant, Library, Lakeside, Auditorium, and Golf course. This classification has practical applications in automated tagging, image search optimization, and content recommendation systems. We employ both supervised and semi-supervised decision trees for this task.

Key challenges include handling data imbalance, extracting informative features, managing model complexity, preventing overfitting, and comprehensively evaluating performance. To address these, we implement Exploratory Data Analysis (EDA), grayscale conversion, Principal Component Analysis (PCA) and Cross Validation to effectively classify venue images.

2. Proposed Methodology

Our dataset comprises a total of 3926 color images (BGR), categorized into 5 distinct classes. The images are in the JPEG format, with a resolution of 256 x 256 pixels.

Fig 1 shows the histogram of all 3 channels for each class. Clearly the images intensities are very similar, only 3 classes are distinct in blue channel [1] [2] [3] [4] [5] [6](golf course, library, lakeside). Green channel shows a slightly skewed histogram across restaurant and auditorium class. Red channel shows a normal distribution where only auditorium class has different intensities.

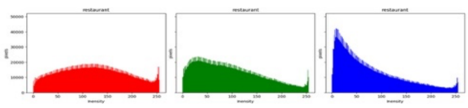

Figure 1: Histogram of restaurant

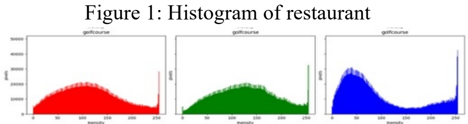

Figure 2: Histogram of Golf course

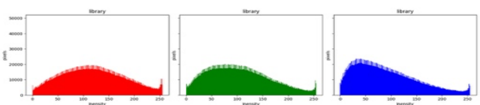

Figure 3:Histogram of Library

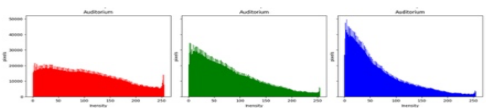

Figure 4: Histogram of Auditorium

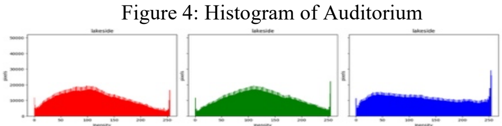

Figure 5: Histogram of Lakeside

Feature Extraction: All the classes are highly overlapped as per the histogram of intensities. In order to have a Decision Boundary between classes we implemented Feature Extraction via PCA.

Principal Component Analysis (PCA) [7]:Integrating PCA into our preprocessing pipeline has proven beneficial for our image recognition model. It not only improved the accuracy and efficiency but also helped in managing the complexity of the data, making it a vital step in our methodology.

Hyperparameters: To optimize the Decision Tree classifier's performance, we tuned several hyperparameters. The key hyperparameters we tuned include maximum depth, minimum sample split, minimum sample leaves and maximum leaf nodes.

Decision Tree (Supervised)

To perform multi-class classification, we used [8]Decision Tree using supervised methodology. We split the input data into Train and Test set to perform training and prediction. PCA is used to perform Feature Extraction before adding the input data. We are training with 3 different settings

- Grayscale input data
- Color input data (BGR)
- Color input data with PCA

All the models are trained with the image size of 64x64 and with the same hyperparameters.

Decision Tree (Semi-supervised)

We implemented s [8]emi-supervised methodology with decision tree to generate pseudo-labels while improving accuracy. Both PCA and Decision Tree are used for training. We splitted the input data into 3 sets Train, Test and Validation. The model is trained and then prediction are made to generate the pseudo-labels on validation dataset. We included the images with predictions having the probability more than 90% into the train dataset. The model is trained is trained for several iterations while increasing the train size and decreasing the validation size.

3. Solving the problem

Supervised Learning

Table 1 shows the test results of Decision Tree trained with labels after a splitting ratio of 70:30. We performed experiments using Grayscale and Color images with and without PCA. Color images with PCA gave us the maximum accuracy 0.56 over the unseen data and we performed hyperparameter optimization with these images

Image size	Decision Tree with Color Images (BGR)	Decision Tree with Grayscale Images	Decision Tree with PCA and Color Images (BGR)
64x64	0.47	**0.42**	**0.56**
96x96	0.50	0.39	0.52
128x128	0.496	0.39	0.53
160x160	**0.512**	0.38	0.51
192x192	0.512	0.40	0.55
224x224	0.512	0.39	0.52
260x260	0.47	0.39	0.52

Table 1: *The accuracy of the supervised Decision Tree model*

Comparison with different methods.
We performed Hyperparameter tuning using **Cross Validation** with the **KFold** of 10. Each hyperparameter is changed over a sequence of iterations while keeping other parameters to Default.
We draw an Average Precision curve for all hyperparameters. The Average precision is found by

$$\textbf{Average Precision} = \frac{1}{N}\sum_{k=0}^{N} (\text{precision})$$

Here, N = 10

Max Depth We changed the Maximum Depth of Decision Tree from 5 to 25 with a step size of 5. Fig shows that the maximum Average Precision is obtained when Max Depth = 10.

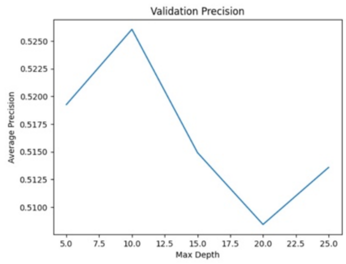

Figure 6: *Hyperparameter Tuning: Max Depth*

K=10	Max Depth = 5	Max Depth = 10	Max Depth = 15	Max Depth = 20	Max Depth = 25
Average Precision	0.519	**0.526**	0.514	0.508	0.513
Average Recall	0.490	**0.518**	0.515	0.512	0.519
Average F1	0.493	**0.515**	0.511	0.505	0.511

Table 2: The *performance metrics of supervised Decision Tree model*

Minimum Nodes Setting the minimum number of nodes more that 300 gave us average precision more than 0.5125.

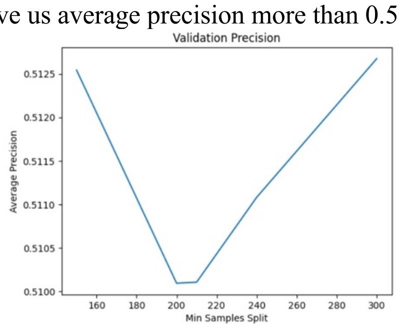

Figure 7: *Hyperparameter Tuning: Minimum Sample Split*

Minimal Samples Leaves Fig 8 shows that the Minimum Samples Leaf parameter around 40 appears to be the most effective based on this graph.

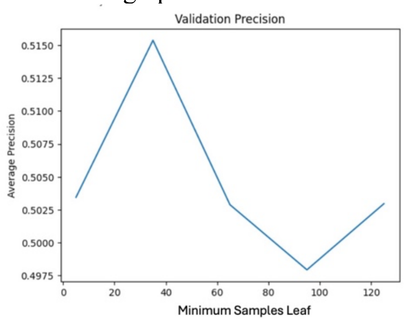

Figure 8: *Hyperparameter Tuning: Minimum Sample Leaves*

Performance
Fig shows the Confusion Matrix obtained after training the Decision Tree with supervised learning. Here, we can note that the class Auditorium gave us maximum **True Positive** (**TP**) while the class Library and Restaurant has maximum **False Positive** (**FP**).

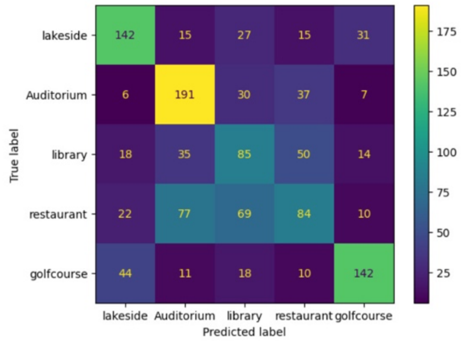

Figure 9: *Confusion Metrix on test set*

The model performed well in identifying Auditorium and Golf course images but had misclassifications between Restaurant and Library, and between Lakeside and Golf course images.

Semi-supervised Learning Model
The semi-supervised learning curve illustrates model performance in terms of accuracy as the training size increases. The blue line represents test accuracy, peaking around a training size of 2000 before slightly declining, indicating the diminishing benefit of additional labeled data beyond this point. The orange line, representing validation accuracy, shows more pronounced fluctuations with significant drops around training sizes of 1200 and 2200, suggesting potential overfitting or instability when incorporating more pseudo-labeled data.

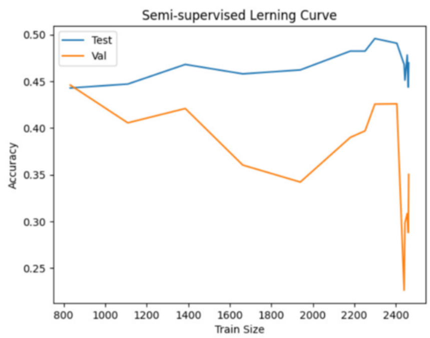

Figure 10: the semi-supervised learning curve

Maximum Depth	Minimum Nodes	Minimum Leaves	Image Added	Iterations	Test Accuracy	Test Samples
5	2	1	663	8	0.43	1281
6	2	1	1097	8	0.45	857
7	2	1	1314	16	0.41	630
8	2	1	1671	12	0.45	273
8	20	1	1609	12	0.47	335
8	40	1	1349	12	0.45	595
8	25	10	1586	11	0.44	358

This table presents the findings of a semi-supervised model, detailing various parameters and their impact on the model's performance.

Challenges
- Too much image overlap reduces the overall accuracy and gave high False Positive FP.
- Model is Overfitting very early over the training dataset.
- The Test Accuracy stop around 55%.
- Hyperparameter Tuning has no effect of model performance.

4. Future Improvements

I. We tried several options and optimizations to improve the overall accuracy, but the high overlap of data points limits model learning. We would like to try the following options to get better results

- Data Augmentation to improve training data.
- Non-Linear dimensionality Reduction techniques like ISOMAP and KernalPCA.
- Random Forest or XGBoost models for image classification.

5. Reference

[1] " https://www.keboola.com/blog/pca-machine-learning#:~:text=When%20PCA%20is%20used%20as,data%20and%20omits%20the%20noise.," [Online].

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

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

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

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

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

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

[8] "https://www.quora.com/Why-do-we-convert-RGB-to-Gray-image-in-pre-processing," [Online].

[9] "https://www.ibm.com/topics/decision-trees#:~:text=A%20decision%20tree%20is%20a,internal%20nodes%20and%20leaf%20nodes.," [Online].

[10] "https://www.ibm.com/topics/semi-supervised-learning#:~:text=Semi%2Dsupervised%20learning%20is%20a,for%20classification%20and%20regression%20tasks.," [Online].