First Phase (Decision Trees)

Mohammad Soroush Ghorbanimehr
40282337
sorush.gm@gmail.com

Fahimeh Nezhadalinaei
40222652
Fahime.nezhadali@gmail.com

Akhil
40306047
akhilxd1@gmail.com

1. Introduction and Problem Statement

In this project, we implemented decision tree models using both supervised[1] and semi-supervised[2] methods to classify the scenes dataset[3] into five classes: bakery, bowling, airport, classroom, and music studio. The objective is to construct a model that can accurately identify these scene classes. This classification task holds significant potential for applications such as automated image tagging and scene recognition.

A major challenge in this project is the selection and tuning of hyperparameters, which have a great impact on the model's performance. The difficulty lies in finding a balance between model complexity and generalization. Additionally, it should be noted that we had to replace two of our original classes—school and hospital—because the dataset was mostly included the exterior images of these categories.

2. Proposed Methodologies

For our supervised and semi-supervised decision tree models, we used the same scene dataset and classes. We used 500 images from each of the five classes of the dataset. The supervised model was trained on a dataset[2] consisting of 1750 (70%) samples, with 375 (15%) samples allocated for testing and an additional 375 (15%) samples reserved for evaluation purposes, bringing the total dataset size to 2500 samples. Additionally, a set of hyperparameters is used to explore different configurations, including criteria such as Gini impurity and entropy, along with varying depths and minimum samples for node splitting. These hyperparameters were specifically chosen to conduct thorough testing and evaluation of the model's performance across different scenarios.

Similarly, the semi-supervised model implemented with a similar set of hyperparameters to assess its efficacy in leveraging both labeled and unlabeled data for classification. By incorporating semi-supervised learning techniques, we aimed to show the model's ability to generalize and make predictions on unseen instances more effectively. Additionally, in the semi-supervised approach, the dataset comprised 20% labeled data, amounting to 350 images, and 80% unlabeled data, equal to 1400 images. Similar to the supervised method, 350 images were allocated for each of the test and evaluation sets, ensuring consistency in the evaluation process.

Throughout the experimentation process, careful consideration was given to the selection and fine-tuning of hyperparameters, recognizing their critical role in shaping the model's performance. By systematically varying parameters such as criterion, max depth, and min samples split, we aimed to identify the optimal configuration that maximized classification accuracy and F1 score. This meticulous approach allowed us to gain deeper insights into the behavior of both supervised and semi-supervised decision tree models, facilitating informed decisions regarding their suitability for scene classification tasks.

3. Solving the problem

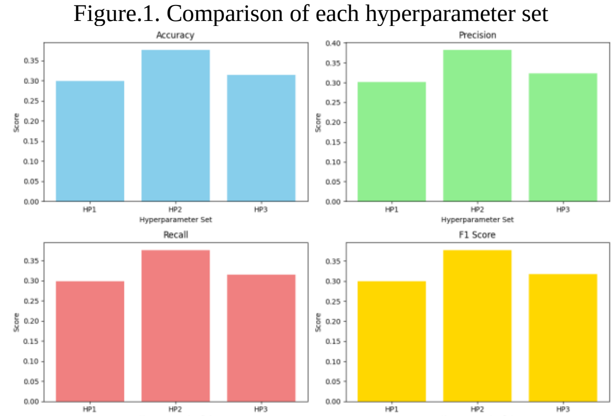
3.1. Decision Tree

For the supervised decision tree classifier, we experimented with different hyperparameters to find the optimal configuration. We used three sets of hyperparameters, and the results varied significantly. First, we used Gini as the criterion, with a max depth of 10 and a minimum samples split of 2. The confusion matrix showed a scattered distribution of correct and incorrect predictions across all classes. Next, we experimented with using Entropy as the criterion, a max depth of 15, and a minimum samples split of 4. This set of hyperparameters provided the best performance among all our trials. Lastly, we reverted to Gini as the criterion but increased the max depth to 20 and the minimum samples split to 5. Despite the deeper tree, the performance did not surpass the Entropy-based configuration. The results of each hyperparameter set are shown in Tabel.1.

Hyperparameters	Accuracy	Precision	Recall	F1 Score
C= Gini, M = 10, Mi = 2	0.2987	0.3009	0.2987	0.2992
C= Entropy, M = 15, Mi = 4	0.376	0.3822	0.376	0.3777
C= Gini, M = 20, Mi = 5	0.3147	0.3228	0.3147	0.3176

Tabel.1. Performance Metrics based on different hyperparameter set

Figure.1. Comparison of each hyperparameter set



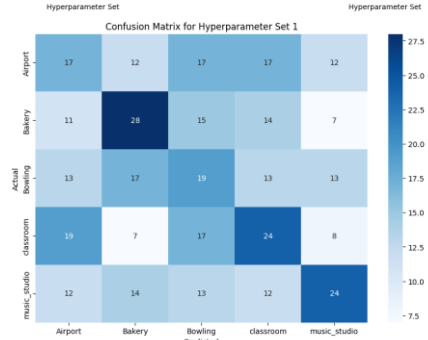
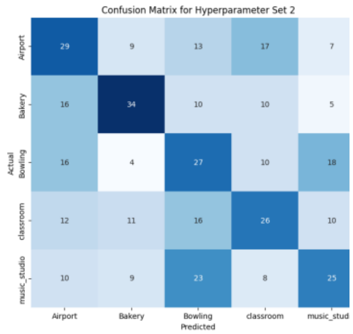


Figure.2. Confusion matrix hyperparameter set 1



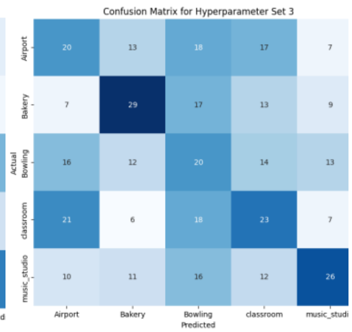


Figure.3. Confusion matrix hyperparameter set 2

Figure.4. Confusion matrix hyperparameter set 3

3.2. Semi-Supervised Decision Tree

For the semi-supervised decision tree classifier, we employed a self-training approach where the model was initially trained on a small portion of labeled data. The model then predicted labels for the unlabeled data, adding the most confident predictions to the labeled dataset. This iterative process aimed to improve the model's performance. Using Gini as the criterion with a max depth of 10 and a minimum samples split of 2, the semi-supervised model achieved an accuracy[4] of 30.4%, with a precision of 30.70%, recall of 30.4%, and an F1 score of 30.19%. Despite multiple iterations, the accuracy did not significantly improve. Using Entropy as the criterion with a max depth of 15 and a minimum samples split of 4, the model achieved an accuracy of 27.73%, with a precision of 28.58%, recall of 27.73%, and an F1 score of 27.87%. This configuration did not perform as well as the Gini-based approach. Lastly, using Gini as the criterion with a max depth of 20 and a minimum samples split of 5, the model achieved an accuracy of 29.07%, with a precision of 29.43%, recall of 29.07%, and an F1 score of 28.85%. Again, the performance was not substantially better. When applying different confidence thresholds (0.75, 0.85, and 0.95) for pseudo-labeling, the performance remained consistent, with the best accuracy being 30.4%. We also tried an iterative approach where we added confident predictions to the labeled set in multiple iterations. Despite adding around 1400 confident predictions in each of the five iterations, the performance of the models did not significantly improve, with the highest accuracy reaching only 30.4%. The results are presented in Tabel. 2.

Hyperparameters	Accuracy	Precision	Recall	F1 Score
C= Gini, M = 10, Mi = 2	0.304	0.3070	0.304	0.3019
C= Entropy, M = 15, Mi = 4	0.2773	0.2858	0.2773	0.2787
C= Gini, M = 20, Mi = 5	0.2907	0.2943	0.2907	0.2885
threshold': 0.75	0.304	0.3070	0.304	0.3019
threshold': 0.85	0.304	0.3070	0.304	0.3019
threshold': 0.95	0.304	0.3070	0.304	0.3019
C= Gini, M = 10, Mi = 2 - ITERATIVE	0.304	0.3070	0.304	0.3019
C= Entropy, M = 15, Mi = 4 - ITERATIVE	0.2773	0.2858	0.2773	0.2787
C= Gini, M = 20, Mi = 5 - ITERATIVE	0.2907	0.2943	0.2907	0.2885

Tabel.2. Performance Metrics based on different hyperparameter set

Overall, our best performing model was given by the supervised decision tree with the Entropy criterion, achieving an accuracy of 37.6%. The semi-supervised approaches did not yield significant improvements, likely due to the high variability and complexity of the image data.

Figure.5. Confusion Matrix of first three Hyperparametes

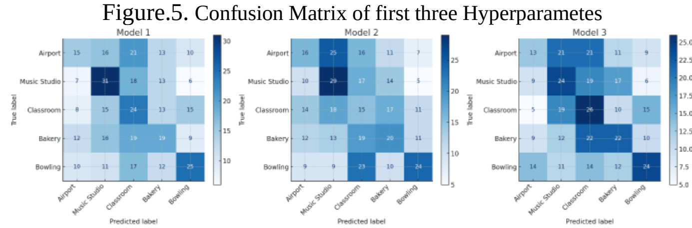


Figure.6. Confusion Matrix of three Thresholds

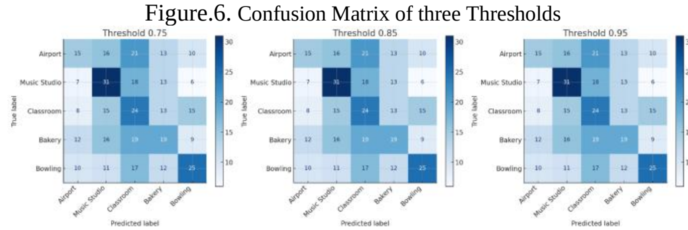
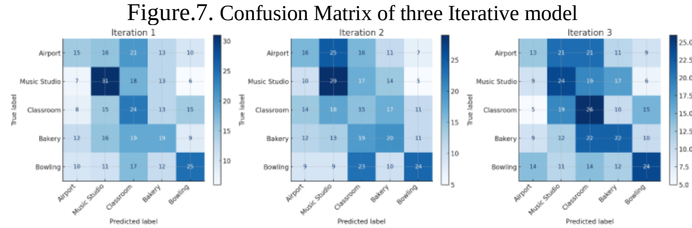


Figure.7. Confusion Matrix of three Iterative model



4. Future Improvements

To improve the accuracy of our model, we plan to train a Convolutional Neural Network (CNN) on this dataset. CNNs are known for their superior performance in image classification tasks due to their ability to automatically extract and learn hierarchical features from images. By leveraging CNNs, we aim to address the limitations observed with the decision tree classifiers, particularly in handling high-dimensional image data and capturing complex patterns. Our approach will include:

1. Model Architecture: Designing a CNN architecture suitable for scene classification, potentially starting with established models like ResNet or VGG and fine-tuning them for our specific task.
2. Data Augmentation: Applying various data augmentation techniques such as rotation, flipping, and scaling to increase the diversity of our training data, thereby improving the model's robustness.
3. Hyperparameter Tuning: Conducting extensive hyperparameter tuning to find the optimal settings for learning rate, batch size, number of layers, and other parameters.
4. Transfer Learning: Utilizing pre-trained models on large-scale image datasets and fine-tuning them on our dataset to leverage the learned features.

By incorporating these strategies, we expect to significantly enhance the model's accuracy and overall performance in classifying the scenes into the specified categories.

References

1. https://link.springer.com/chapter/10.1007/0-387-25465-X_9
2. https://scikit-learn.org/stable/modules/generated/sklearn.semi_supervised.SelfTrainingClassifier.html
3. http://places2.csail.mit.edu/download.html
4. https://medium.com/analytics-vidhya/accuracy-vs-f1-score-6258237beca2