COMP 6721 Phase 1 Report

Group - ResNet

Vibhor Gulati
40238072

Dhyey Nilesh Doshi
40244534

Namrata Pankajkumar Brahmbhatt
40233323

I. Introduction and Problem Statement

With data being generated in a huge amount there needs to be a process which helps us figure out the relevance and more use cases of it. One way of figuring out the relevance of the data in different circumstances is by labelling it appropriately so we know which data can be used in which context and make decisions backed by statistics of the operations performed on the data. Unfortunately, the amount of data being generated is not small and hence labeling is not an easy task and there are often privacy restrictions as well which prevent data from being labeled at the source where it is generated. This leads to an abundance of unlabeled data which is of little significance till we label it correctly.

In this project we aim to explore solutions to label a set of unlabeled images so that they can be further used for other purposes. There are a few researched techniques which help us classify the data by training a model on the available labelled data and predicting the labels for the rest. We also aim to improve prediction by tuning different hyperparameters.

One of the most widely used techniques for image classification is supervised learning and semi-supervised learning. We will be performing these learning techniques using decision trees on a partition of the set of images and then try to label the remaining set in 5 different classes.

Major challenges faced during the implementation were gathering suitable data sets to be used for training and testing and improving the prediction by the model. After managing to find the datasets, another challenge was removing the outliers and selecting which subset of images to use. The filtered dataset had another issue which was that the depth of images was not consistent and hence we faced issues running our code on them.

Limited data was another major issue since we need to train our model under a controlled environment and since image data is high-dimensional it also needs a lot of feature space compared to textual data.

We considered writing python scripts to filter out the images of different depth. We also investigated dimensionality reduction techniques to reduce the feature space.

II. Proposed Methodologies

i. Dataset and Preprocessing

The dataset comprises images from data sources [1][2][3][4][5] for five different classes: Bar, Casino, Restaurant, Library, and Hospital. The images are loaded and preprocessed using three steps.

Firstly, images are loaded from a specified directory. During this process, images not in RGB format are filtered out, ensuring consistency in the input data. Secondly, each image is resized to 256x256 pixels to standardize the input dimensions. Lastly, pixel values are normalized to the range [0, 1] by dividing by 255.0.

ii. Feature Extraction

The 256x256x3 images are flattened into 1D arrays of size 196,608. According to Culurciello, et al. [6] transforming 3D into a series of 1D arrays applied across the channels resulting in an approximate 2 times increase in evaluation speed compared to the standard model. The flattened model maintains or even surpasses the accuracy of traditional models, while utilizing only one-tenth of the parameters.

Principal Component Analysis (PCA) according to Greenacre et al. [7] is a flexible statistical technique used to refine data into its key elements, known as principal components. This method allows for an approximate reconstruction of the original data table using only these significant components. It even further reduces the number of features, significantly lowering the computational cost and potentially improving model performance.

iii. Model

a) Supervised Learning

Decision Tree classifier is chosen for its simplicity and interpretability. The classifier is implemented using Scikit-learn's DecisionTreeClassifier. The model's hyperparameters try to prevent overfitting and enhance generalization.

Hyperparameter tuning is done using a manual Grid Search for parameter optimization[8]. This method simplifies the optimization process by automatically testing each parameter combination, thereby saving time and effort. It evaluates the model's performance across the parameter grid, helping to identify the optimal combination that enhances the model's performance. This makes the tuning process more efficient and reduces the likelihood of human error. The three parameter we tuned are:

max_depth: Controls the maximum depth of the tree. Prevents overfitting by limiting the complexity of the model.

min_samples_split: The minimum number of samples required to split an internal node. Balances the trade-off between depth and breadth of the tree.

ccp_alpha: Complexity parameter for Minimal Cost-Complexity Pruning. Helps in reducing the size of the tree by

pruning nodes that do not provide a significant gain in performance.

b) Semi-Supervised Learning

In real-world scenarios, getting labeled data for machine learning models is very costly. Hence it is difficult to use supervised learning in the real world. Hence semi-supervised learning is used when dealing with such types of datasets.

Initially, semi-supervised learning starts with a limited labeled dataset to train an initial model and uses this model to infer labels for the unlabeled data. The confidence of each prediction is assessed, which generally uses the prediction probabilities. High-confidence predictions are then chosen based on predefined thresholds. These high-prediction pseudo-labeled instances and then combined with the original labeled data, creating an expanded labeled dataset.

The model is then retrained using this expanded dataset. This process of pseudo-labeling, high-confidence selection, and combining the data is performed iteratively. This process continues until the maximum number of iterations is reached.

Hyperparameter tuning for semi-supervised is performed like that of supervised learning with the same parameters being tuned.

iv. Model Evaluation

The model's performance is evaluated using accuracy, precision, recall, F1-score, and confusion matrix. These metrics provide a comprehensive view of the model's strengths and weaknesses across different classes.

III. SOLVING THE PROBLEM

i. Failed Attempts

The model was unable to train on the dataset initially due to different bit depth of the images.

Training the Decision Tree with default parameters resulted in low accuracy (~20-30%). The model overfitted the training data due to high dimensionality and lack of regularization.

Additionally, attempting to train the model on the flattened images without dimensionality reduction led to the model struggling to generalize due to the vast number of features.

Semi-supervised learning focused on iterations and threshold without using Scikit-learn's SelfTrainingClassifier gave low accuracy (~25%)

ii. Successful Attempts

Applying Principal Component Analysis (PCA) for dimensionality reduction significantly improved the training speed by reducing the feature space.

Hyperparameter tuning was systematically conducted using grid search, which identified the optimal settings for the model. This tuning process was crucial in balancing the model's complexity and performance. Although the best model had ccp_alpha set to 0.0, experimenting with pruning parameters revealed its potential for future enhancements.

For semi-supervised learning, 20% of training data is randomly selected as labeled data, the rest of the data is treated as unlabeled data and the labels we have in the dataset are ignored and set as -1. The decision tree model is trained on the labeled data. The decision tree model predicts the labels for the unlabeled data using the SelfTrainingClassifier and the confidence of this prediction is calculated using the same.

SelfTrainingClassifier for semi-supervised learning provided better accuracy with k-best criterion.

iii. Results

For supervised learning we were able to obtain accuracy of 38.5%. Bar had the highest precision of 0.52 and Casino had the highest recall with 0.61. The f1-score was observed around 0.45 for 3 classes.

For semi-supervised learning, an accuracy of 31% was obtained which is lesser than that obtained for supervised but it is an expected behavior. Casino and Hospital had comparatively higher precision with 0.44 and 0.43 respectively. Similar to supervised learning, casino had the best recall for semi-supervised as well. An f1-score of about 0.42 was observed for 2 classes.

IV. FUTURE IMPROVEMENTS

A bigger data set can help improve the results of the predictions for both supervised and semi-supervised training.

More experiments can be done with hyperparameters, and they can be better fine-tuned.

More sophisticated techniques can be explored for semi-supervised learning.

Instead of decision trees, other models like Convolutional networks can get better results for image classification.

V. REFERENCES

[1] Mittal, S. Places, Version 10. Retrieved June 1, 2024 from https://www.kaggle.com/datasets/mittalshubham/images256/version/10

[2] Ahmad, M. MIT Indoor Scenes, Retrieved June 1, 2024 from https://www.kaggle.com/datasets/itsahmad/indoor-scenes-cvpr-2019/data

[3] a Joson, N. Places-2_MIT_Dataset, Version 2, Retrieved June 1, 2024 from https://www.kaggle.com/datasets/nickj26/places2-mit-dataset/version/2

[4] Anh, V. Restaurants, Retrieved June 3, 2024 from https://www.kaggle.com/datasets/airbornbird88/restaurants

[5] images.cv | Labeled image datasets, Retrieved June 1, 2024 from https://images.cv

[6] Jin, Jonghoon, Aysegul Dundar, and Eugenio Culurciello. "Flattened convolutional neural networks for feedforward acceleration." arXiv preprint arXiv:1412.5474 (2014)

[7] Greenacre, M., Groenen, P. J., Hastie, T., d'Enza, A. I.,

Markos, A., & Tuzhilina, E. (2022). Principal component analysis. Nature Reviews Methods Primers, 2(1), 100.

[8] M. Hammad Hassan, Using Grid Search For Hyper-Parameter Tuning (2023) https://medium.com/@hammad.ai/using-grid-search-for-hyper-parameter-tuning-bad6756324cc (Last accessed June 6, 2024)