Team Non-ECE gators: Brand Logo Classification

Nima Ajayebi
s.ajayebi@ufl.edu

Ahmid Bin Kabir Rabbi
a.rabbi@ufl.edu

Ali Katooziani
al.katooziani@ufl.edu

***Abstract*—The project that was carried out by the Non-ECE gators was to develop a brand logo classification model. To achieve this goal, various machine learning models were tested from simple models to more advanced complicated models. The first step was to utilize support vector machine (SVM) and naïve bayes (NB) classifiers. Even though they were helpful to achieve insight to decide between using HiPerGator or personal laptops, the model's accuracy was not acceptable. Even after implementing principal component analysis (PCA) the accuracy didn't go beyond 47 percent. In the next step, convolutional neural networks (CNNs) were tested to achieve better results. This included testing different layers for the network, adding dropouts and regularizers. Yet, the resulting accuracy which was 59 percent wasn't satisfactory. After some exploration, the team concluded that employing transfer learning was the most suitable approach for image classification, and hence the optimal choice for this particular undertaking. This included utilizing different pre-trained models, unfreezing different number of layers and adding regularizers to the model. All these efforts lead to a satisfactory accuracy of 90 percent. Overall, this project helped the team to implement the methods that were taught in the class, in a project to have a better understanding of those methods.**

I. Introduction

Logo classification has become increasingly significant due to its various uses, such as identifying copyright violations, suggesting products, and displaying context-based advertisements. Real-world logo images are more diverse in appearance and have more intricate backgrounds compared to other types of object images. As a result, identifying logos from images is difficult. Logo classification has several practical applications, including identifying counterfeit products, automating quality control, and improving search and retrieval systems [1].

Transfer learning is a type of machine learning technique that allows a model trained on one task to be reused as the starting point for a different but related task. In traditional machine learning, a model is trained from scratch on a large dataset, and then the model's weights are fine-tuned for a specific task. This process can be computationally expensive and requires a significant amount of data to achieve high accuracy. Transfer learning, on the other hand, takes advantage of the knowledge and patterns learned by a model during its training on a large dataset, and applies this knowledge to a new task. The pre-trained model is often a deep neural network that has learned to recognize complex patterns and features in images, text, or audio. The pre-trained model's weights are then frozen, and additional layers are added on top to adapt the model to the new task. These additional layers can be trained using a smaller dataset, which reduces the amount of data and computational resources required. In the context of logo classification, transfer learning involves using a pre-trained model on a large dataset of images to extract meaningful features and patterns from logos. These features are then used to train a new model specifically for logo classification [2].

This approach can significantly reduce the amount of training data and computational resources required to build an accurate logo classification model. This approach to logo classification can help businesses create a more efficient and accurate visual identity management system, enabling them to make data-driven decisions about their branding and marketing strategies [3].

In this project some of machine learning classification models; such as; support vector machine (SVM) and naïve bayes (NB) are first trained on the dataset. Then after evaluating the results, a convolutional neural network (CNN) is built to classify the logos. At the end, the task to classify the logos is carried out using the transfer learning.

II. Implementation

The first step was the data collection stage. This stage consisted of collecting images from 10 different logos and putting them in a dataset. Combining all the images that was collected by the students created the dataset of this project.

The next step involved manually reviewing and reassigning any incorrectly labeled entries or images to their appropriate categories. This was a crucial step as having a human verify the labels ensured that the data's true values were consistent and minimized errors or challenges during the training and validation phases of the process.

The third step in this project was to use SVM and NB to start classifying the logos. The first machine learning classification that was tested on the dataset was SVM. SVM is a machine learning algorithm that works by finding a

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

hyperplane in a high-dimensional space that optimally separates the different classes of data points. Naive Bayes classification is a probabilistic machine learning algorithm that uses Bayes' theorem to predict the class of a given data point based on the probability of each class given the features of the input. It assumes that all features are independent, hence the term "naive". Other than the results, there was some merits using these models. The team was trying to understand if any of these models would have any meaningful results to begin the project. To achieve this purpose, the team used the same code that was developed in homework 3.

The next step was to train a CNN model which showed to be capable of image classification tasks. Instead of preprocessing the data to derive features like textures and shapes, a CNN takes just the image's raw pixel data as input and "learns" how to extract these features, and ultimately infer what object they constitute. Different parameters and layer numbers were tested to achieve good accuracy. To avoid Overfitting some neurons dropped in each section by Dropout function and also an L2 regularizer was used.

The final step was using transfer learning in this project. Transfer learning is a widely adopted technique in machine learning that reuses knowledge from pre-trained models to boost the performance of new tasks with limited data. By exploiting shared structures and features between source and target tasks, transfer learning enables the effective re-application of pre-trained models in diverse scenarios. In this study, we employed transfer learning to classify images of ten classes, experimenting with three pre-trained architectures: MobileNetV2, Xception, and ResNet50V2. These models have been pre-trained on the extensive ImageNet dataset.

MobileNetV2 is a lightweight model designed for efficient inference on mobile and edge devices, using dep-thwise separable convolutions and inverted residual structure to reduce complexity while retaining performance.

Xception employs depth-wise separable convolutions but introduces a distinct architecture based on mapping spatial and channel-wise correlations independently, enabling more complex representations with fewer parameters.

ResNet50V2 is an enhanced version of the original ResNet architecture, introducing residual connections for deeper models without the vanishing gradient problem. Pre-activation residual units and updated weight initialization improve accuracy and convergence speed.

These experiments aimed to identify the most suitable architecture for the specific dataset, showcasing the efficacy of transfer learning in adapting to novel classification tasks and highlighting its potential for further research and practical applications.

III. Experiment

The results of the different models and different steps that were used for this project can be found as follow:

A. *Data Pre-processing*

While working with the dataset, the team come to notice that some of the images are misclassified. Correcting these misclassified images helped to increase the accuracy of the model. Furthermore, the team used principal component analysis (PCA) for SVM and NB to reduce the complexity of the developed model.

B. *SVM*

In the experimentation phase, the initial algorithm implemented yielded an accuracy score of **0.41** (without PCA) and **0.47** (with PCA) during testing. It can be seen that the accuracy of this model is unsatisfying, therefore, the team started working on a naïve bayes model. During the experimental phase, the team determined the most effective approach to execute programs based on the available resources, such as HiPerGator, personal laptops, and UF lab computers. To optimize the use of HiPerGator, the team downsized the images in the dataset to ensure they did not surpass the maximum memory allocated for the course's student development environment. This resizing process also contributed to shorter computation times when testing various algorithms.

C. *NB*

The next algorithm that was used in this project was naïve bayes. The results indicated a **0.29** (without PCA) and, a **0.35** (without PCA) accuracy.

D. *Comparing the results of NB and SVM*

Table 1 illustrates the accuracy of the SVM and NB models.

TABLE I. Accuracy of SVM and NB.

Model name	***Accuracy with PCA***	***Accuracy without PCA***
SVM	0.47	0.41
NB	0.35	0.29

While the SVM had the better performance out of the two models, it's clear that none of the options produced satisfactory results. This led the team to explore more intricate models using neural networks, which demonstrated potential for enhancing the classification outcomes.

E. *CNN*

The team initially utilized a straightforward CNN design taught in the lecture and experimented with adjusting model parameters to improve accuracy. Given the large image dataset, the team added more convolution layers, with Dropout function applied in each layer to prevent overfitting. The team tested five different dropout ratios, ultimately choosing 0.4 for the best result. The team also experimented with varying the number of layers in each convolution section from 32 to 256, ultimately selecting the best-performing model with 3 convolution sections and 10 output layers with SoftMax function to match the number of classes.

To determine the optimal number of epochs, the team tested a range from 10 to 100, achieving the best accuracy with 40 epochs. Despite using Dropout, the team still observed overfitting in the model, leading to apply l1 and l2 regularizers. After testing both regularizers, l2 regularizer was selected with the l parameter of 0.01, validated through K-fold (K =5) cross-validation.

While using Dropout improved accuracy in the validation set from **0.48** to **0.59**, it still fell short of the accuracy in training and indicated overfitting. Therefore, the team decided to explore pre-trained CNNs to address the model's lack of generalization and unacceptable accuracy.

F. Trrannsfer learning

Our team conducted an extensive investigation using three pre-trained models to ascertain the most appropriate architecture for the given dataset. We chose a large-sized model (ResNet50V2) comprising 25.6 million parameters, a medium-sized model (Xception) encompassing 22.9 million parameters, and a relatively compact model (MobileNetV2) with a mere 3.5 million parameters. Our primary objective was to evaluate the performance of these diverse models with varying parameter sizes.

Initially, all models encountered overfitting issues. To mitigate these challenges, we applied consistent hyperparameter tuning across all selected models. Within the architecture of the three models, we employed techniques such as regularization and dropout layers, in conjunction with various learning rate values. Moreover, we unfroze a select number of layers in the base model and trained them concurrently with the newly added layers, leading to enhanced model performance. Importantly, we utilized a smaller learning rate to prevent damaging the pre-trained weights.

For performance evaluation, we experimented by unfreezing different numbers of layers while maintaining the same hyperparameters. In our analysis, we unfroze three, five, and seven layers, respectively, to assess the impact on model performance.

TABLE II. ACCURACY DIFFERENT MODELS.

Model Name	***Frozen Layers***	***HP tuned w Frozen Layers***	***HP tuned w Unfrozen Layers= 3***	***HP tuned w Unfrozen Layers= 5***	***HP tuned w Unfrozen Layers= 10***
ResNet50V2	Valid acc: 88% Train acc: 97%	Valid acc: 87% Train acc: 95%	Valid acc: 83% Train acc: 94%	Valid acc: 90% Train acc: 98%	Valid acc: 83% Train acc: 88
Xception	Valid acc: 88% Train acc: 95%	Valid acc: 84% Train acc: 97%	Valid acc: 85% Train acc: 95%	Valid acc: 95% Train acc: 82%	Valid acc: 86% Train acc: 94
MobileNetV2	Valid acc: 84% Train acc: 91%	Valid acc: 86% Train acc: 94%	Valid acc: 86% Train acc: 93%	Valid acc: 99% Train acc: 86%	Valid acc: 84% Train acc: 93%

Based on the results that can be seen in the table 2 the ResNet50V2 with 5 unfrozen layers has the best performance with the 90 percent validation accuracy.

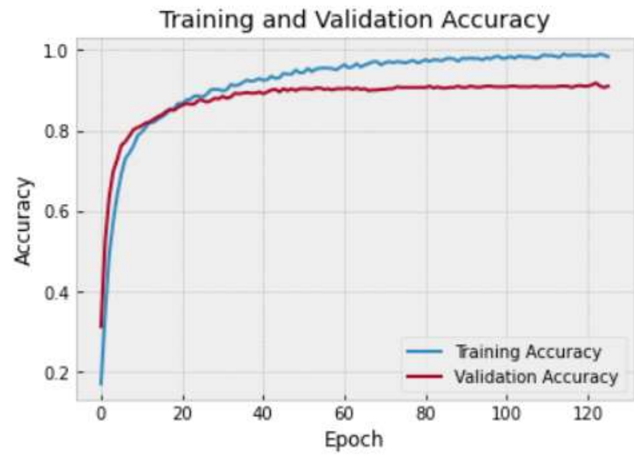

Fig. 1. Training and validation accuracy of the ResNet50V2 with 5 unfrozen layers.

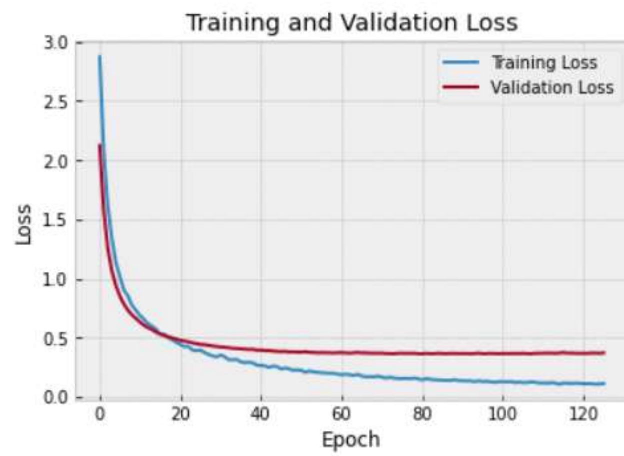

Fig. 2. Training and validation loss of the ResNet50V2 with 5 unfrozen layers.

Figures 1 and 2 present the accuracy and loss results of the training and the validation of the ResNet50V2 with 5 unfrozen layers model for each epoch.

IV. CONCLUSION

Our team was tasked with creating a model that could classify brand logos. We utilized the theory taught in our course to implement different techniques and algorithms for building a machine learning model. Our research revealed that the most effective approach for image classification is to use convolutional neural networks (CNN) and take advantage of transfer learning, particularly by utilizing pre-trained CNN models.

After testing and experimenting with different architectures, we found that our CNN model based on a pre-trained existing model achieved the highest accuracy rate of **90%** in classifying images. This is a significant improvement compared to the other models we tested. We believe that our experience with this project has not only improved our technical skills but also deepened our understanding of machine learning applications and the importance of choosing the right algorithm for a specific scenario.

REFERENCES

[1] J. Wang, W. Min, S. Hou, S. Ma, Y. Zheng, H. Wang, S. Jiang, Logo-2K+: A large-scale logo dataset for scalable logo classification, Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 6194-6201.

[2] L. Torrey, J. Shavlik, Transfer learning, Handbook of research on machine learning applications and trends: algorithms, methods, and techniques, IGI global, 2010, pp. 242-264.

[3] M. Shaha, M. Pawar, Transfer learning for image classification, 2018 second international conference on electronics, communication and aerospace technology (ICECA), IEEE, 2018, pp. 656-660.