Nocturna: Emblem Classifier

David Langus Rodriguez[1], Jorge Luis Alberto[2], and Rebecca G. Hart[3]

***Abstract*—In this paper, a novel Convolutional Neural Network (CNN) architecture is proposed to combat major difficulties such as limited availability of data, overfitting, and computational complexity which are inherent to complex classification problems. The proposed approach utilizes a toy example of logo classification to evaluate the performance of the learning algorithm. The experimental framework employs an image dataset collected by the EEL4773/EEL5840 class and composed of 5933 total images. Techniques such as data augmentation, dropout, and transfer learning are applied to a test set composed of 80% of the images in the dataset while the remaining images were set aside for validation. Experimental results show that a CNN with complex data augmentation, the application of dropout, and transfer learning with EfficientNetB7 achieved an impressive 97.47% accuracy in correctly classifying the data on the validation set.**

I. Introduction

Advancements in computer vision and artificial intelligence have revolutionized the way we interact with visual data. Image classification and object detection has been a major theme within this domain due to it's wide variety of applications such as autonomous vehicles, surveillance systems, and medical applications [1]–[3]. One common application of image classification in machine learning is logo detection. This task can become increasingly complex as logos can vary throughout the years for the same brand as well as the pictures themselves having various settings, scales, illumination, and resolution [4].

To solve these issues, several learning methods have been proposed for image classification methods including Naive Bayes Classifiers, support vector machines (SVMs), and traditional neural network models [5]. However, these methods require manual feature selection which can increase the engineering cost and lower the overall efficiency of the algorithm. One method to reduce the need for feature selection is to use a Convolutional Neural Network (CNN) which automatically select features and have a lower engineering cost [5]. Therefore, they are used frequently in tasks such as image recognition, computer vision, and other applications which use data formatted in a grid [6].

CNNs are composed of components including sets of convolutional layers, pooling layers, and fully connected layers which each help improve the performance of the learning algorithms [7]. The convolutional layers are what distinguish CNNs from other types of neural networks. These layers add filters to the input data where each filter is responsible for detecting various features within the data through using a convolution. The output of this operation is then passed through an activation function to allow the weight associated with that feature to be scaled. Commonly, nonlinear activation functions are used such as sigmoid, hyperbolic tangent, rectified linear unit (ReLU) and the corresponding leaky ReLU. The pooling layer helps reduce the dimension of the feature map which is output from the convolutional layer, this is a key element in reducing the computational cost associated with running the algorithm as well as having the added benefit of reducing the chance of the algorithm overfitting. The final layer of a CNN is the fully connected layer which is most similar to traditional neural network architectures. The output of the convolutional and pooling layers is feed into one or multiple fully connected layers where the final layer in the CNN is responsible for making predictions.

The objective of this report is to propose techniques that will address some of the major difficulties which comes from complex classification problems such as limited availability of data, overfitting, and computational complexity. These issues are addressed using methods such as data-augmentation, dropout, and transfer learning and measured by the success of the model to correctly classify logos from a set of 10 classes.

II. Implementation

In this section we will outline the method in which the data was collected, pre-processed, and labeled as well as the process in which the features were extracted, the training and evaluation of the model as well as additional fine-tuning and iteration. The final component of this section will outline the deployment of the model on an unseen test set.

A. Data Collection

A collaborative effort within the EEL5840/EEE4773 - Machine Learning class resulted in 5933 images of logos for: Nike, Adidas, Ford, Honda, General Mills, Unilever, McDonald's, KFC, the University of Florida Gator, and 3M. Each logo class was encoded using integer encoding as described in Table I. These images (seen in Figure 1) capture the phenomenon of brand logos undergoing change throughout the years and add additional complexity through introducing various backgrounds and lighting conditions between samples.

To enhance the efficacy of the learning algorithm, a series of technical pre-processing steps were carried out. First, each image underwent a cropping and resizing operation, adhering to a user-specified dimension of a 300×300 pixel RGB image.

[1]Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. Email: dlangus@ufl.edu

[2]Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. Email: jorgealberto@ufl.edu

[3]Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL, USA. Email: rebecca.hart@ufl.edu

Table I
INTEGER ENCODING FOR BRAND LOGOS

Brand Logo	Integer Encoding
Nike	0
Adidas	1
Ford	2
Honda	3
General Mills	4
Unilever	5
McDonald's	6
KFC	7
Gator	8
3M	9

Figure 1. Example of Gathered Data

B. CNN Model Architecture

The architecture of the developed CNN is organized into three distinct blocks, each characterized by a single convolutional layer followed by a subsequent MaxPooling layer. Utilizing zero padding and a nonlinear activation function, these blocks effectively capture intricate non-linear relationships within the data. The first second and third convolutional layers are composed of 16, 32, and 64 filters respectively. Upon the completion of the final block, the model integrates a potential dropout layer (further detailed in Section II-C2) and subsequently flattens the output. This flattened representation is then channeled into a fully connected layer, each consisting of 128 neurons equipped with nonlinear activation functions which feeds into the output layer composed of 10 neurons.

C. Methodology

After initializing the CNN model, a series of procedures were implemented to improve the classification abilities of the algorithm while attempting to mitigate overfitting. The iteration included augmenting the data through two different augmentation processes. Additionally, a dropout procedure was implemented for both augmented and not augmented datasets. As a final step, transfer learning was added to the model to improve performance.

1) Data Augmentation: Data augmentation is a widely adopted methodology employed to artificially augment the dataset's size, serving to enhance the model's generalization capabilities. This approach encompasses the application of diverse transformations to the images, encompassing random rotations, flips, zooming, and alterations in brightness, thus enabling the model to adapt to these variations during training, leading to improved generalization performance.

To conduct a detailed comparative analysis of the impact of data augmentation, two distinct models were employed. The first augmentation process, termed "Basic Data Augmentation," involved subjecting images to a series of random transformations. This process included random horizontal flipping, random rotation within a range of ± 0.1 radians, and random zooming with a scale of 1 ± 0.1.

The second augmentation process, referred to as "Advanced Data Augmentation," was similar in that it applied horizontal flipping and rotations to the input images. However, it also flipped the images vertically and the rotational range was increased to ± 0.5 radians, and the zoom factor was expanded to 1 ± 0.5. Additionally, to address variations in lighting conditions, contrast adjustments were made within a range of ± 0.5 relative to the original image's contrast value. This process was then repeated with the image being flipped horizontally, a rotation with a range of ± 0.2 radians, contrast being adjusted from ± 0.1 of the previously adjusted value, and a random zoom with a factor of 1 ± 0.2 of the previously augmented value.

2) Dropout: Dropout is a regularization technique applied in convolutional neural networks (CNNs), involving the random deactivation of specific neurons during training. This method aims to enhance the model's robustness against overfitting by reducing the reliance on particular features. Consequently, the CNN's generalization capacity is improved, leading to enhanced performance on previously unseen data.

During the training process, a dropout layer was introduced which would stochastically deactivate a percentage of the neurons in the final `MaxPooling2D()`layer, for each model the dropout rate was selected to be 0.2 indicating that 20% of the neurons would be deactivated. The selected parameters can be found in Table II.

3) Transfer Learning: Transfer Learning offers significant benefits within the realm of machine learning by leveraging the knowledge gained from models trained on large-scale datasets to enhance model performance with limited training data. For image identification tasks, models can be fine tuned using transfer learning from models that specialize in recognizing specific patters. This method is particularly valuable when data is difficult to obtain or computationally expensive to process. Three transfer learning processes were preformed using popular convolutional neural network architectures: MobileNetV2, EfficientNetB0, and EfficientNetB7.

MobileNetV2 was selected due to it's ability to balance the size, speed, and accuracy of the model. This model allowed for the efficient and streamlined integration of transfer learning into our model. EfficientNetB0 is employed as an intermediary transfer learning mechanism, aimed at facilitating the exploration of hyperparameters for the more intricate EfficientNetB7. The incorporation of EfficientNetB0 serves to mitigate computational burdens during preliminary computations, streamlining the subsequent tuning phase. After the hyperparameters were selected, the model was adjusted to accept EfficientNetB7 which has one of the highest top 1 and top 5 accuracy scores of the models available in the $\sim$ /.`keras`/`models`/repository.

III. Experiments

Before experimentation, the training data was pre-processed using the method described in Section II-A. Upon completion of the pre-preprocessing stage, the training data was partitioned into an 80-20 ratio, with 20% of the data being set aside as a validation set for the model after training.

Experiments were carried out by training multiple configurations of the CNN utilizing combinations of Data Augmentation, Dropout, and Transfer Learning. For comparison, a preliminary run will be conducted using the convolutional neural network described in Section II-B without Data Augmentation, Dropout, or Transfer Learning. For the initial experimentation, each training set was constructed by running the model for 10 Epochs and a batch size of 42. After initial experimentation was conducted, the number of epochs was increased to improve the estimation capabilities of the algorithm. The developed models were then evaluated on the validation set and the performance was tabulated in Table II.

Our experimental process was comprised of a systematic exploration of various techniques which were theorized to enhance the performance of the CNN. Initially, we evaluated the CNN without any enhancements and then introduced isolated instances of data augmentation, described in section II-C1, and dropout techniques. The use of data augmentation and dropout was then combined with transfer learning with the final focus being on different available transfer learning models, outlined in II-C3, and their impact on the logo-classification problem. Because of the number of hyperparameters available in the composite framework of data augmentation, dropout, and transfer learning the `keras_tuner.Hyperband()`algorithm was used subsequent to the selection of other pertinent parameters and the initial assessment of transfer learning's effectiveness. This strategic approach aimed to harness an extra performance boost.

A. Results and Discussion

Each model configuration from Table II the model was trained using the same train-test split and all models except the EfficientNetB7 transfer learning case used ReLU activation functions. For Trial 4.6 and 5.6 the SELU activation function was selected. At each epoch, the model accuracy was evaluated using a validation set and the training and validation set accuracy were recorded. Each proposed improvement yielded an improvement in the overall performance of the CNN. Particularly noteworthy is the improvement achieved through the integration of Basic Data Augmentation and the dropout method. This combination yielded an impressive 10.89% improved prediction accuracy on the validation test set compared to the standalone developed CNN.

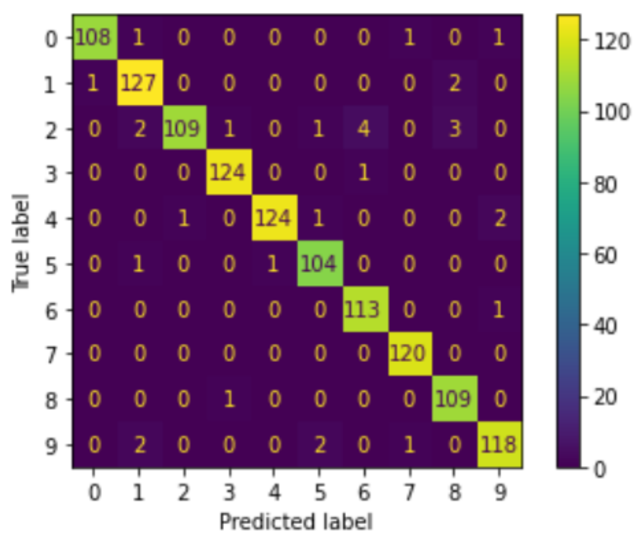
However, the most significant increase in performance came from employing transfer learning with the worst preforming transfer learning algorithm, integrated with the dropout and data augmentation methods, was the MobileNetV2 transfer learning algorithm with both basic and advanced Data Augmentation. Nevertheless, the percent increase compared to the original CNN was 33.73% and 36.89% respectively. The best preforming transfer learning algorithm was EfficientNetB7 which had correct classification accuracy of 94.10% on the validation set when using Advanced Data Augmentation. The addition of `keras_tuner.Hyperband()` for hyperparameter tuning allowed this model to reach 97.47% correctly classified data on the validation set yielding a improvement of 50.30% over the original CNN's performance.

The confusion matrix shown in Figure 2 demonstrated that our developed algorithm had the most difficulty classifying class 2, Ford, often misclassifying it as McDonald's or the

Table II
Optimization Methods and Their Impact on Classification Accuracy

Trial	**Optimization**			**Accuracy(%)**	
	Dropout	Transfer Learning	Dat. Aug.	Val.	Train
1.1	-	-	-	64.87	98.57
2.1	-	-	Basic	71.86	77.56
2.2	-	-	Adv.	63.10	61.42
2.3	✓	-	-	62.43	95.89
3.1	✓	-	Basic	71.95	79.84
3.2	✓	-	Adv.	63.61	64.98
4.1	✓	MobileNetV2	Basic	86.69	92.08
4.2	✓	MobileNetV2	Adv.	88.80	90.29
4.3	✓	EfficientNetB0	Basic	93.85	98.10
4.4	✓	EfficientNetB0	Adv.	92.25	94.40
4.5	✓	EfficientNetB7	Basic	93.09	96.80
4.6	✓	EfficientNetB7	Adv.	94.10	98.53
5.1	✓	MobileNetV2 w/ Hyperband	Basic	92.42	99.39
5.2	✓	MobileNetV2 w/ Hyperband	Adv.	92.50	97.62
5.3	✓	EfficientNetB1 w/ Hyperband	Basic	94.61	99.37
5.4	✓	EfficientNetB1 w/ Hyperband	Adv.	94.69	97.47
5.5	✓	EfficientNetB7 w/ Hyperband	Basic	96.88	99.89
5.6	✓	EfficientNetB7 w/ Hyperband	Adv.	97.47	99.77

Figure 2. Confusion Matrix of Trial 5.6



UF Gator Logo. This could be due to the presence of a mustang and many curved lines within all the logos as well as significant variations in lighting which were noted when the data was displayed and plotted during the model development and can be seen in Figure 1. It had the least difficulty with class 3,6,8, and 7 (Honda, McDonalds, the UF Gator, and KFC) the latter having no incorrect classifications while the first 3 only had one incorrect prediction for each class.

IV. Extra Credit

The supplementary credit assignment entailed modifying the machine learning algorithm to handle instances where an image outside the original classification set outlined in Table I is introduced. To fulfill this task, we implemented a mechanism that, upon encountering an unclassified image, assigns a label of -1, signifying its divergence from the known classes. The subsequent changes documented in this report illustrate the intricate adjustments devised to enable this functionality.

To make the discrimination between images which fall outside the categories outlined in Table I a distribution of the probabilities used to classify the samples was plotted and used to develop a threshold function. This function allows a user defined input in the form of a percentage which will classify an image as unknown, and consequently labeled as -1, if the probability that the image belongs to any of the classes in Table I falls below the user defined input.

V. Conclusions

In this study, we conducted a series of comprehensive experiments on Convolutional Neural Network (CNN) architectures targeting prevalent image classification challenges, including data-set size, overfitting, and computational efficiency. By employing various combinations of advanced techniques such as Data Augmentation, Dropout, and Transfer Learning, we sought to identify an optimal architecture for the logo-detection algorithm toy example. After a series of experiments were conducted, the final model which integrated Data Augmentation, dropout method, and transfer learning with EfficientNetB7 and achieved a classification accuracy of 97.47% on the previously unseen validation set.

References

[1] A. Juyal, S. Sharma, and P. Matta, "Deep Learning Methods for Object Detection in Autonomous Vehicles," in *International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 751–755, June 2021.

[2] V. Tsakanikas and T. Dagiuklas, "Video surveillance systems-current status and future trends," *Comput. Electr. Eng.*, vol. 70, pp. 736–753, Aug. 2018.

[3] L. Cai, J. Gao, and D. Zhao, "A review of the application of deep learning in medical image classification and segmentation," *Ann. Transl. Med.*, vol. 8, p. 713, June 2020.

[4] J. Sanghvi, J. Rathod, S. Nemade, H. Panchal, and A. Pavate, "Logo detection using machine learning algorithm : A survey," in *Int. Conf. Comput. Commun. Appl.*, pp. 136–141, Mar. 2023.

[5] S. Nath, G. Mishra, J. Kar, S. Chakraborty, and N. Dey, "A survey of image classification methods and techniques," in *Int. Conf. Commun. Inf. Comput. Technol.*, pp. 554–557, July 2014.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[7] S. Krishna and H. Kalluri, "Deep learning and transfer learning approaches for image classification," *Int J Recent Technol Eng*, vol. 7, no. 5, 2019.