Image Classification and Object Detection with Artificial Neural Networks

Alan Smith
Dept. of Mechanical and Aerospace Engineering
University of Florida
Gainesville, Florida
smithal6092@ufl.edu

***Abstract*—This paper explores the effectiveness of Artificial Neural Networks and transfer learning on classification and detection tasks. Two tasks were presented: classifying images of flowers by species and detecting cars within images of a road. Both tasks were solved via Convolutional Neural Networks with transfer learning. The flower datset utilized the Xception CNN model and obtained an accuracy in test of 90%. The car dataset utilized two MobileNetV2 models; one for classification and one for objection detection. The classification model performed extremely well, obtaining 93% in training and showing very accurate results for test. The detection model exhibited poor performance in training the model, given high validation loss, however, exhibited good results on the test set. The value in implementing Artificial Neural Networks and transfer learning, given the appropriate understanding and computational resources, was made evident though this project.**

***Index Terms*—artificial neural network, transfer learning, image classification, object detection**

Fig. 1. Left: Image of 'Rose' flower species. Right: Image of car with car detection bounding box.

I. Introduction

Artificial Neural Networks (ANN) are a subset of Machine Learning (ML) and represent an essential concept for generating high-performing ML models. Imitating the human brain, where information is passed via a connection of neurons, ANNs are comprised of aggregated layers of weighted neurons for passing and transforming input data. The input data is passed along a network of layers, each performing a unique task, until a desired output label, i.e. regression value or classification label, is attained. For modifying the input data, activation functions (a function for transforming some input value) are regularly implemented within layers to obtain an appropriate output. Also, gradient descent is utilized for minimizing a loss function for optimizing weights associated with each neuron. There are various of types of ANNs, however, all offer benefits such as high performance, parallel processing, and extreme model control. This paper evaluates the performance of artificial neural networks as applied to image classification for species of flowers and detecting cars (Figure 1).

One of the most effective neural networks utilized in solving image classification tasks is the Convolutional Neural Network (CNN). CNNs are networks comprised of three primary layers: the convolutional, pooling, and the fully-connected (FC) layer. The convolutional layer features a nxn matrix which translates across the image performing element wise matrix multiplication. This convolution represents a filter, extracting information from the image, such as obtaining definitive edges or blurring the image. Next a pooling layer is implemented which down samples the incoming image i.e. reduces its resolution. Pooling also utilizes nxn matrices to translate across the pixels and, depending on the approach (max pooling, average pooling, etc.) will extract and output the optimal pixel value. At the cost of information loss, computational efficiency drastically improves. Finally, the FC layer simply connects all remaining image information to the output neurons to perform classification or regression.

An efficient approach to training artificial neural networks is through transfer learning. Transfer learning is a technique which uses pre-trainined models to perform new but related tasks. For example, numerous image classification networks are accessible for extracting and applying parameters to new image-based problems. Learned weights from the pre-trained model are transferred to new input data and implemented. As a result, computation time to train a new model is drastically reduced as reliable weights have already been computed. Additionally, transfer learning alleviates the amount of training data required as reliable models would have been trained with sufficient data. Modifications, however, are necessary. Depending on the transferred model's structure, it is likely that the input and output network layers of the data itself need to be modified in order to effectively utilize the model. Additionally, to fine-tune a model, additional layers can be added/removed or the layer weights can be re-calculated to better fit the new data. Regardless of the approach, transfer learning is incredibly useful in optimizing models or establishing benchmarks. This project utilizes two learned models. First, the 2016 'Xception' model which uniquely implements depthwise separable convolution layers, layers which separately model spatial and cross-channel (RGB) patterns. Second, the MobileNetV2 model, an algorithm designed for efficiency with mobile devices and one which bypasses bottlenecks between layers.

ANNs were implemented on images of flower species and cars in the interest of optimizing classification performance and efficiency. ML models were optimized and fit to processed data in order to gain valuable insight into the approach necessary when implementing ANNs.

II. Approach

300x300x3 images of flower species and 380x676x3 images of cars were processed and classified. The steps taken in establishing and fittings the complete models are detailed.

A. Data Preprocessing

To accommodate the associated transfer learning model, all original flower species data was reshaped from 270,000 features to 300x300x3 and later scaled with a MinMaxScaler function. No further pre-processing was necessary for the data.

The car detection dataset was rehaped into 380x876x3 images and scaled with a MinMaxScaler.

For both tasks, a stratified split(80/20) was implemented on the training data for generating training and validation data during CNN learning.

B. Define Models

The flower species data utilized transfer learning and implemented the Xception model for image classification. Excluding the input and output layers, the base Xception layers and weights were retained. Two models were used in fitting the training data. First, the base Xception model was used for establishing a benchmark. For this model, no layer weights were retrained. The second model allowed layer weights to be trained, along with additional hyperparameters which are to be discussed. For the tuned model, a batch normalization layer was added as this technique accelerates learning while improving generalization. Both models used a sparse categorical cross entropy loss function and Adam optimizer due to the limited parameters necessary for tuning.

The car detection problem was subdivided into two parts. First, a CNN for classification and, second, a modified CNN for car detection for appropriate samples. Both models incorporated transfer learning with the MobileNetV2 model. For the classification model, layer weights were made trainable, and the output was a binary value of 0 or 1 to represent that a car is not present (0) or vice versa (1). The output layer utilized a softmax activation function and was compiled with the sparse categorical cross entropy function. The detection model also made weights trainable, however, had four output values, utilizing the softplus activation function to compute bounding box locations. For this model, the loss function used the mean squared error as this was a regression task. It is noted, for the detection stage, only the training data featuring cars was utilized and, similarly, this model only made predictions on test data that had been classified with a '1'.

C. Design Tuning Approach

Hyperparameter tuning was implemented for optimizing all defined models. While evaluating more hyperparameters increases computation time, the ability to obtain higher performance models is yielded. Due to the extreme computational cost, primarily in system memory, only essential hyperparameters that were considered to make a beneficial contribution to the models were evaluated. Additionally, as the system memory cap was often met and the Python kernel would often crash, a custom Randomized CV Search was implemented to successfully evaluate parameters for the flowers dataset. For car models, the values were input manually as alternate methods exceeded the computational resource limit. The hyperparameter values tested were batch size, learning rate, epochs, and the number of trainable layers (as taken from the transfer learning model). Due to system constraints, a 2-fold CV scoring was ran on the models for determining the optimal parameters for each model.

D. Model Implementation and Evaluation

All models were fit to the training data and tuned with the validation data, retaining the hyperparameter values that generated the best performance. The models were then ran on the associated test sets and their performance was evaluated according to their associated metrics.

III. Results

Various ANN algorithms were implemented on test data from pre-processed images. Their performance in classifying various images and detecting objects is detailed.

A. Flower Species Dataset

Transfer learning via the 2016 Xception model was implemented to classify flower images by species. Following hyperparameter tuning, the optimial hyperparameters were determined to be a batch size of 32, learning rate of 0.0001 and 100 for the Xception layer to begin re-training weights. The learning curve for the training and validation sets during model fitting is shown below in Figure 2. The resulting accuracy and associated confusion matrix for the training and test sets are shown in Table I and Figure. 3, respectively.

TABLE I
Flower Species Classification

	Training	**Test**
Accuracy	98.0%	90.0%

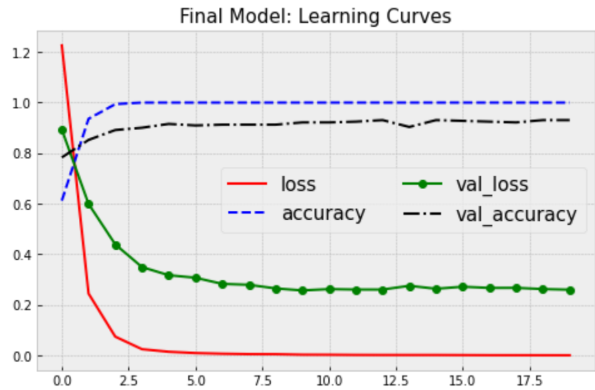

Fig. 2. Learning Curve of tuned model for flower species classification.

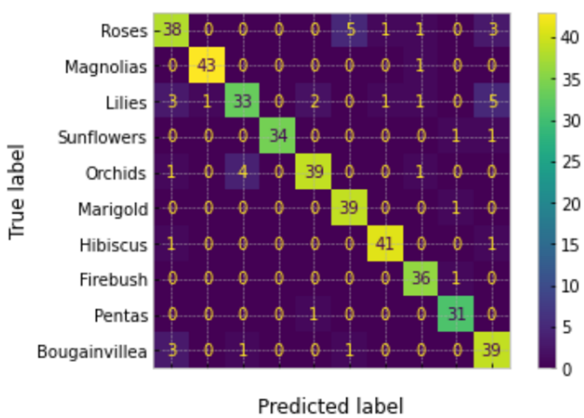

Fig. 3. Confusion Matrix for flower species test set.

Fig. 4. Images of test images as classified by the MobileNetV2 CNN classifier. The image on the left is correctly classified, the image on the right is incorrectly classified.

B. Car Classification and Detection

Two transfer learning models, utilizing the MobileNetV2 model, were implemented to perform car classification then detection. For classification, the optimal hyperparameters were determined to be a bathc size of 32, a learning rate of 0.0001 and the re-training began at layer 224 of the transfer model. The training set resulted in an accuracy of 93.0%. As training labels were not provided for the test set, a definitive accuracy score was not readily calculable, however, Figure 4 denotes a good representation of performance. From a visual inspection, the majority of test images were classified correctly, however, Figure 4, highlights instances where challenging samples (samples where a car is small or partially out of frame) were correctly and incorrectly classified. For the detection model, the optimal hyperparameters were determined to be a batch size of 32, learning rate of 0.0001, and 75 epochs. The learning curves for each model are shown below in Figure 5 and 6. Output images from the test set are shown in Figure 7. Despite a good test performance, detection in training was found to be extremely inconsistent, with the majority of boundary boxes surrounding incorrect locations.

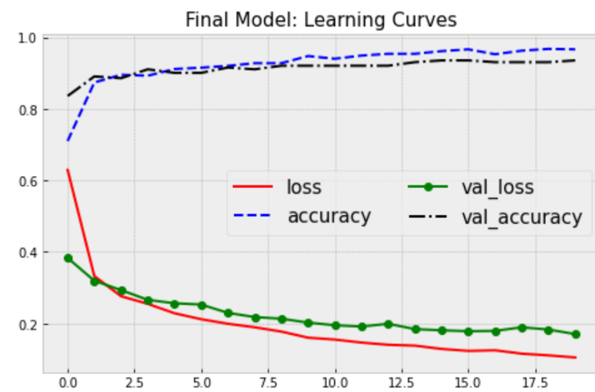

Fig. 5. Learning curve of tuned model for car classification.

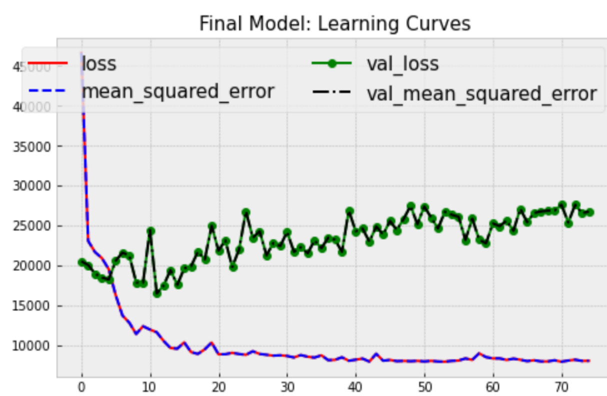

Fig. 6. Learning curve of tuned model for car detection.

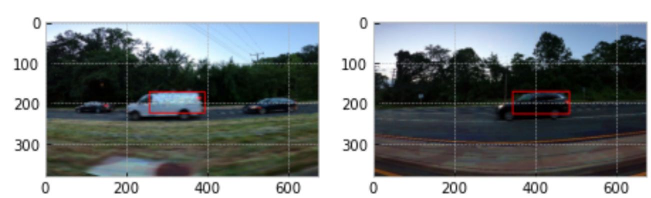

Fig. 7. Example images from car test set after car classification and detection.

Discussion and Conclusion

Images of flower species and road cameras were processed through various transfer learning models in order to gain insight into the impact and effectiveness of ANNs.

The initial and apparent insight given the flower species task is the extreme effectiveness and benefit in utilizing transfer learning. As shown, using the standard Xception model, with no parameter tuning or modification to layer weights, a classification accuracy of 87.0% was obtained in validation. To an experienced ML practitioner, room for improvement is evident and attainable. However, for many new users, or those lacking resources, understanding, or time, the ability to utilize a pre-existing model and still obtain a very high performance is available. Upon tuning the model, accuracy in training was 98% and 90% in test. This shows, the model likely overfit the training data, however, 90% accuracy was still a high performance. In modifying the models, the benefit in training layer weights was highlighted, however, only to a certain point. Hyperparameter tuning revealed that re-training a large number of layers increased computation time while failing to significantly improve performance. Therefore, defining where layer weights should be trainable is an essential parameter to optimize.

Transfer learning also demonstrated sufficient effectiveness for the car detection task. As stated, this task was subdivided into two tasks, first classification, then detection. Utilizing the MobileNetV2 model, the classification model showed excellent results, obtaining a 93% score in training and performing extremely well, visually, on the test set. The majority of images were classified correctly, with inaccuracies occurring on challenging samples, shown on the right in Figure 4. Utilizing a detection model on the test samples (those classified as having a car present), bounding boxes were found to be fairly accurate as well. Shown in Figure 7, boxes were effective in finding the car and locating the bulk of its body, however, smaller sections of the car (the hood) were often excluded. For a quantitative performance measure, a mean average precision (maP) can be computed for the trained bounding box, assuming a ground truth box can be defined for the test samples. It is crucial to note, however, the poor performance in training for the detection model, justified by the poor learning curve performance in Figure 6. As shown in the learning curve, the model overfit the training data as seen by the increasing validation loss. When the final model was implemented on the complete training dataset, the bounding boxes failed to detect properly, as expected. As discussed, however, the test set exhibited great performance. Consequently, the approach is clearly effective, however, further tuning and modifications are necessary in order to generate a reliable model.

Despite the evident advantages in utilizing ANNs, this project highlighted the delicate trade-off between model performance and computational cost. The models implemented for these tasks were performed with a A100 GPU, a extremely high end GPU, and 16 GB of memory. Due to the cost, these are computational resources that are generally not easily accessible. Despite this, however, the models implemented regularly crashed the Python kernel, for a variety of code scripts, and exceeded the allocated memory. This played a major factor in hyperparameter tuning and simplified the process into a randomized CV approach with only 2 CV folds or manual tuning. While models could generally be implemented, an ideal tuning approach (grid search with more parameters) was severely restricted. Additionally, regardless of the memory allocations, running the CNN models locally on a personal computer was significantly slower or not possible. Consequently, neural networks are not suitable for all tasks given the extreme computational cost required to accommodate them.

ANNs with transfer learning proved to be extremely effective methods for image classification and object detection. Implementing proven, pre-trained CNN models demonstrated high accuracy with minimal or no tuning. Specifically, minimal tuning was effective in bringing classification performance to 90.0% accuracy or higher. Conversely, utilizing the same models for object detection, and modifying for a regression task, proved to be partially effective. Tuning the model failed to achieve low validation loss, however, implementing the model on the test set still demonstrated good results. Finally, the trade-off between performance and computational cost was highlighted through these tasks. Regardless, ANNs are truly valuable algorithms for obtaining high performance on classification and regression tasks, as shown. The value in utilizing transfer learning was evident given the simplicity in establishing a model, the training efficiency, and the high performance given a base model. Given the proper understanding and resources, ANNs are an essential tool for solving a variety of machine learning tasks.