Flower Classification and Object Detection using Artificial Neural Networks

Akash Kumar Kondaparthi
Dept. of Electrical and Computer Engineering
University of Florida
Gainesville, FL, USA
akash.kondaparth@ufl.edu

***Abstract*—This paper presents a comprehensive study of Deep Learning techniques applied to two datasets in particular: one for image classification, and another for object detection. In the realm of machine learning and artificial intelligence, the capacity to correctly classify and detect objects within imagery data serves as a pivotal benchmark for the advancement of practical applications. This project delves into the application of deep learning techniques for the classification of flower species and the detection of vehicles within images. Utilizing a robust dataset comprising 1,678 images across 10 flower species, an artificial neural network was trained, yielding a remarkable classification accuracy of 99% for the training set and 88% for both validation and test sets. Furthermore, the employment of a transfer learning approach with ResNet50 architecture mirrored these results, cementing the efficacy of the model. The exploration continued with the implementation of a state-of-the-art object detection algorithm, YOLOv8, on a dataset annotated with 559 car bounding boxes. The model demonstrated exceptional precision, as evidenced by a Mean Average Precision (mAP50) score of 0.988 for both training and validation datasets. These findings underscore the models' capabilities in handling diverse tasks within the visual domain deployment in various business and research-oriented applications.**

I. Introduction

The automation of image-based classification and detection tasks has become a cornerstone challenge in machine learning, with significant implications for sectors ranging from automotive safety to agricultural technology. This project addresses the problem of accurately classifying various species of flowers and detecting cars in different settings using artificial neural networks, a task that simulates real-world scenarios where visual recognition plays a crucial role.

The datasets employed in this study include a collection of 1,678 flower images spanning 10 distinct species, each represented as a 300×300×3 RGB image, and a set of 559 images annotated with vehicle bounding boxes for object detection. The flower dataset provides a balanced representation of classes, while the car dataset presents a typical use-case scenario in automated surveillance and traffic management systems. The methodology adopted involves training deep learning models: a custom neural network for flower classification and the cutting-edge YOLOv8 for car detection, each fine-tuned through rigorous hyperparameter optimization to achieve optimal performance on both training and unseen data. The successful application of these models showcases the potential of convolutional neural networks in processing and making predictive decisions based on visual data.

II. Methodology

A. Data Preprocessing

For the flower species classification, the dataset images were normalized, which scaled the RGB values to a [0,1] range to facilitate faster convergence during training. In the car detection dataset, only the images with at least a car present in the image were considered (with bounding box labels). Images without labels were discarded due to computational and temporal restriction of this project.

B. Model Training and Validation

For the flower classification task, two neural networks were experimented with: a custom neural network and transfer learning with ResNet50. The neural network from scratch was implemented by fine tuning the hyperparameters so that the validation accuracy improved without overfitting. The initial neural network to start with outputted the learning curves as shown in Figure 1. There are some obvious signs of overfitting since the learning curves for training and validation do not seem to be much corelated. Hence, more convolutional layers were added to extract more features, number of dense layers were reduced and methods such as adding or removing Convolutional or Dense layers, adding regularization techniques such as Dropout layers and Pooling layers, experimenting with the complexity of the model based on the learning curves were carried out. For Transfer Learning with ResNet50, few of the top layers (top 5 layers) were only allowed to learn. The rest of the layers were frozen to maintain the model's 'imagenet' weights.

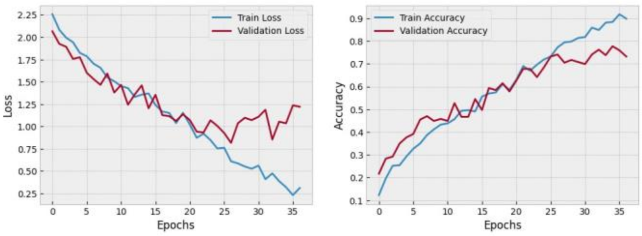

Figure 1. Initial Performance of the neural Network from scratch

In the object detection task, the YOLOv8 architecture was utilized to detect cars on highways. The algorithm was trained to recognize the spatial features within the images that correspond to vehicles and to predict the bounding boxes around them. A pre-trained network was used on this dataset and fine-tuned it to take advantage of transfer learning, which significantly improved learning efficiency and performance.

C. Model Evaluation

The evaluation of the models used different metrics since these models are purposed for different applications. The flower classification models used standard metrics: accuracy, precision, recall, F1 score, and AUC-ROC for the classification task. For object detection, the YOLO model used mean Average Precision (mAP) and Intersection over Union (IoU). These metrics provided with a comprehensive assessment of the models' performance.

The learning curves of loss and accuracy over epochs for both training and validation sets were also visualized, which helped monitor and adjust the training process dynamically.

III. Results

A. *Neural Network from Scratch*

The first model, a custom neural network, on the flower classification dataset achieved a training accuracy of 99%, with validation and test accuracies of 88% and 86%, respectively after a lengthy process of hyperparameter tuning. The model architecture was settled on with 2,167,434 parameters with five convolutional layers and three dense layers. The model outputted the following learning curves as shown in Figure 2, and the following confusion matrices for Training and Testing respectively as shown in figures 3 and 4.

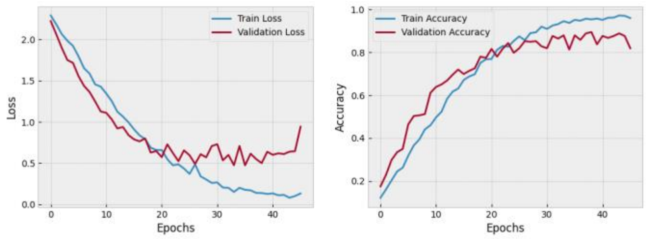

Figure 2. Neural Network from scratch learning curves.

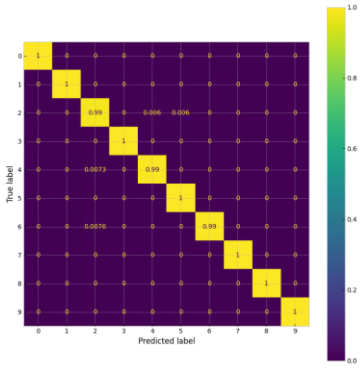

Figure 3. Neural Network from scratch confusion matrix on the training set.

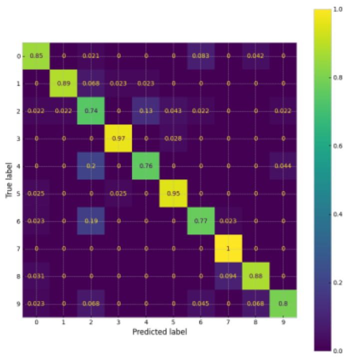

Figure 4. Neural network from scratch confusion matrix on the test dataset.

B. *Transfer Learning with ResNet50*

The second approach, which utilized transfer learning with the ResNet50 architecture, mirrored the training accuracy of the first model and improved both validation and test accuracies to 88%. The learning curves showed an excellent fit, with a steady decrease in loss and a plateau in accuracy, indicating that the models learned effectively without overfitting. The learning curves and the confusion matrices on training and test sets are shown in figures 5 through 7.

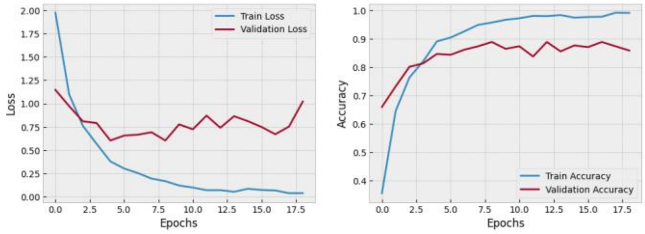

Figure 5. ResNet50 learning curves.

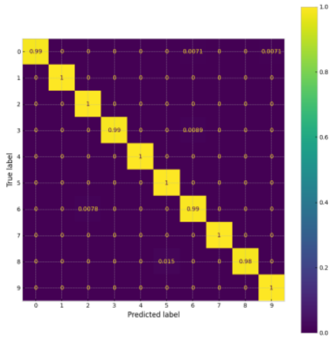

Figure 6. ResNet50 confusion matrix on the training set

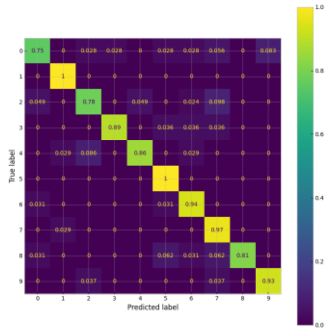

Figure 7. ResNet50 confusion matrix on the test set.

The performances of these models are again summarized in Table1.

Performance	Precision	Recall	F1-score	Accuracy
Neural Network from scratch	0.87	0.86	0.86	85.06%
Transfer Learning with ResNet50	0.90	0.89	0.89	89.03%

Table 1. Performance comparison of flower classification models on the test set.

C. *Object Detection with YOLOv8*

The YOLOv8 model was trained for object detection, achieving impressive mAP50 scores of 0.988 for both training and validation datasets. Sample predictions from the test dataset are visually illustrated in figure 8, where the model has confidently localized cars with high probabilities enclosed in bounding boxes. These qualitative results complement the quantitative mAP scores, showcasing the model's robustness in recognizing and locating vehicles across different scenes. Figure 9 shows the learning curves for the YOLOv8 model and figure 10 shows the precision-recall curve for the validation set.

Figure8. Validation examples from the YOLOv8 model.

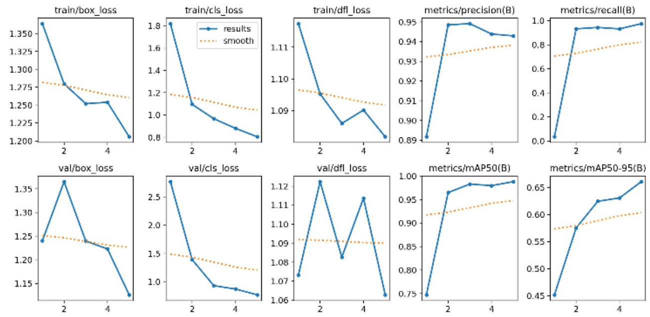

Figure 9. Learning curves for the YOLOv8 model.

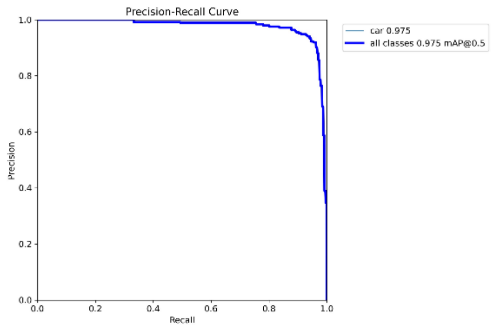

Figure 10. Precision-Recall curve for the YOLOv8 model.

For the car detection task, the bounding box predictions were quantitatively evaluated using IoU and mAP metrics. The mean Average Precision (mAP50) was 0.988 for both training and validation sets.

IV. Discussion

A. *Neural Network from Scratch*

The neural network from scratch showed high accuracy for the number of parameters, mostly due to the benefit of hyperparameter tuning. The overfitting in the neural network was dealt with by adding dropout layers in the dense part of the CNN. The feature size of the images were also set to maximum (300 x 300 pixel images).

The ruggedness of the learning curves can be observed due to lower batch size of the training. Lower batch sizes have rugged learning behaviors and higher batch sizes have smoother curves.

B. *Transfer Learning with ResNet50*

The ResNet 50 model used images which were downsized by half (150 x 150) pixel images. Also, it is apparent from the learning curves of the ResNet50 that the accuracy and performance starts off at a much better value than when using a neural network from scratch. This is expected because we are using transfer learning by implementing ResNet50. This ResNet50 was imported with model weights ‘imagenet’ and hence it has the advantage of learning more datasets beforehand so that it can perform better on the flower classification dataset.

The learning curves for ResNet50 are much smoother than for the neural network from scratch. This is the case because, ResNet50 has lower number of “trainable” parameters, so there is smaller change for every epoch to go with which results in the smoother curves as shown in the figure 5.

C. *Object Detection with YOLOv8*

The YOLOv8 model's success in car detection, as indicated by the high mAP scores, showcases the power of modern object detection frameworks to handle complex tasks like localizing objects within images. The model's ability to detect cars with high confidence across varied backgrounds and lighting conditions speaks to its robustness and practical applicability in real-world scenarios.

Since YOLO is a real-time high-end model trained on a larger number of datasets, this model has a headstart with the performance when it comes to object detection. The proposed strategies for validation, including manual annotation and the use of IoU and mAP metrics, underline the importance of having reliable evaluation methods for deployment readiness.

For the Object Detection task, since there are no target labels provided for the test set, one simple way to validate is by visual inspection. But this process is not optimal for large datasets, so by choosing a subset of dataset and manually giving them labels, we can validate the performance based on the ground truth labels.

In the availability of bounding box labels, to validate the performance on the test set, we can use Mean Average Precision (mAP) and Intersection over Union (IoU) metrics, which take into account the overlapping Regions of Interest (ROIs).

IoU: Intersection over Union measures the overlap between predicted and actual bounding boxes. It's calculated by dividing the area of overlap by the area of union of these boxes. In scenarios where there's an acceptable margin of error in bounding box location, IoU helps determine how 'close' the predictions are to potential true positives. A threshold (like 0.5) is often set, above which a prediction is considered a true positive.

mAP: This metric evaluates the model's precision across different levels of recall, essentially assessing how well the model detects objects (cars, in this case) across varying confidence thresholds. For each confidence level, precision and recall are calculated, and the average precision (AP) is derived. mAP is the mean of these APs across all classes or IoU thresholds. It's crucial when exact bounding box locations aren't the target but rather the model's overall ability to detect cars with a reasonable degree of accuracy.

For images without cars, we can add a fixed target label [0,0,0,0] during training. This teaches the model to recognize 'no car' scenarios. During validation, if the model predicts a bounding box where none should exist, it would negatively impact the mAP, as this would be a false positive.

D. *Implications and Conclusions*

The results achieved by the models have several implications for business applications. In the case of flower species classification, the high accuracy of the models can lead to the development of mobile applications for educational or botanical research purposes, allowing users to identify flower species in real-time. For car detection, the model could be integrated into surveillance systems for traffic monitoring or automated parking solutions.

The discussions and conclusions drawn from this study emphasize the potential of neural networks in image classification and object detection tasks. They also highlight the importance of careful model evaluation and the need to consider the end-use case during model development. The strategies and methodologies discussed in this report provide a roadmap for approaching similar problems in other domains, ensuring that the models are not only statistically sound but also practically viable.

V. Conclusions

In conclusion, the project successfully demonstrates the application of artificial neural networks in the classification of flower species and the detection of vehicles, achieving high accuracy and precision across both tasks. The flower species classification model showed excellent training performance and robust validation results, affirming the effectiveness of neural networks in image recognition tasks. The car detection model, using a YOLOv8 framework, exemplified state-of-the-art object detection capabilities, suitable for real-world applications such as traffic surveillance and autonomous driving systems. The results underline the critical importance of hyperparameter tuning, model evaluation, and validation strategies to ensure the reliability and generalizability of machine learning models. Overall, the findings of this report highlight the transformative potential of deep learning in automating and enhancing image-based classification and detection tasks, paving the way for innovative applications in various industries.

References

[1] https://github.com/ultralytics/ultralytics

[2] https://github.com/UF-AppliedMLSystems-Fall23/Lectures/blob/main/Lecture%2027/Lecture%2027-in-class%20edits.ipynb

[3] https://www.tensorflow.org/install/source_windows