Flower Species Classification and Car Detection Using Deep Learning

Dipayan Saha
ECE Department, University of Florida, Gainesville, FL-32611
dsaha@ufl.edu

***Abstract*—Classification and object detection are two important problems to be solved for developing automatic systems. The evolution of machine learning, especially deep learning, has showcased astounding success in performing these tasks. With that inspiration, this work performs a classification task on a flower species dataset and an object detection task on car detection dataset with the help of deep neural networks. The neural networks are built up from scratch and the concept of image processing has also been applied to perform these tasks.**

***Index Terms*—Deep Learning, Machine Learning, Flower Species Classification, Car Detection**

I. Introduction

WITH the development of computational facilities, deep learning has become the center of concentration in machine learning research. The excellent quality of deep neural networks in extracting features in an end-to-end fashion has minimized the efforts of hand feature engineering and also increased performance to a great extent. The capability of deep learning in solving complicated non-linear problems has inspired researchers to apply such neural networks in diversified applications. Biomedical engineering, transportation system, emotion analysis, agriculture, social media and communication system are a few to name.

In recent years, the use of deep learning in food and agriculture engineering has become a popular practice. One of the common practices in this field is to classify different species of flowers, crops, etc. This is important to develop automatic systems. With this motivation, in this work, we perform a 10-class classification task on a flower species dataset.

Transportation system is another field that has witnessed a growing interest in deep learning for different tasks. Service quality assessment, traffic analysis, and autonomous car development are some of them. Objects like cars, traffic signals, and sign boards are important to be detected in a practical environment to develop a self-driving car, which is the talk of the town in the automobile industry. This inspires this work also perform object detection tasks by drawing the bounding box in the location of the car from images.

The remainder of the report is organized into three sections. Section II explains the problem statement and Section III narrates the dataset. Afterward, Section II describes the proposed methodology. Next, Section V details the model setup, experimentation, and result analysis. Finally, Section VI sums up the observation and concludes the paper.

II. Problem Statement

This work deals with two different tasks. The tasks are described below

- **Task 1** The goal of this task is to perform a 10-class classification of flower species from the RGB images.
- **Task 2** The goal of this task is to detect the position of the car in RGB images. There can be also cases where there is no car in the image.

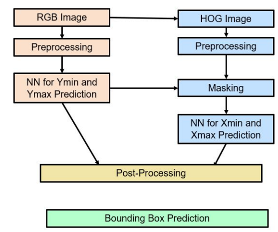

Fig. 1: Proposed methodology for car detection

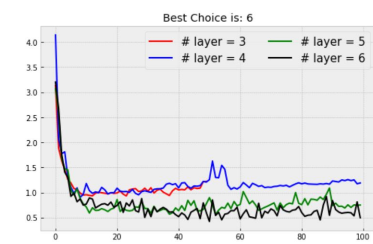

Fig. 2: Learning curves of validation loss for different number of layers in case of flower species classification

III. Dataset

As mentioned before, for performing two machine learning tasks, two different datasets are given. This section describes these datasets in detail.

A. Flower Species Dataset

This dataset contains RGB images of flowers belonging to 10 different species: ‘Roses’, ‘Magnolias’, ‘Lilies’, ‘Sunflowers’, ‘Orchids’, ‘Marigold’, ‘Hibiscus’, ‘Firebush’, ‘Pentas’, ‘Bougainvillea’. There are 1658 images in the training set and 415 images in the test set. The dataset is slightly unbalanced. Each image has a shape of $300 \times 300 \times 3$.

B. Car Detection Dataset

This dataset contains RGB images from two different videos. In this case, the training and test videos are separated by the video in order to perform a proper evaluation of the training set. The dataset is diversified in the sense that some of the images contain a single car, some images contain multiple car, and some images do not have any car. The location of each car is described by 4 labels: Xmin, Xmax, Ymin and Ymax of the bounding box. The original training dataset has around 500 labels of these bounding boxes. Later, I modified the training set by adding labels for no-car images. The label for no car image has been considered as [0,0,0,]. Though test images were provided, there was no label for the test images. Using the ‘Make Sense AI’ software, test labels have been created. In this case, images with multiple cars have been discarded. In total, there are 896 and 164 images in the train and test sets, respectively. Each of the images has a shape of $676 \times 380 \times 3$.

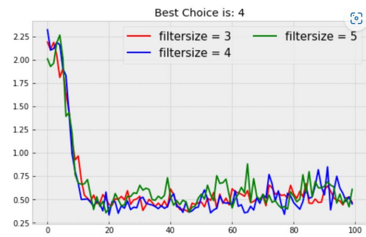

Fig. 3: Learning curves of validation loss for different filter size in case of flower species classification

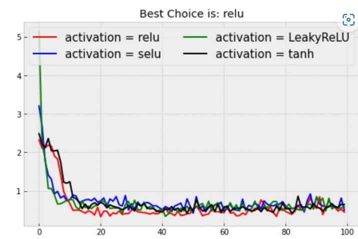

Fig. 4: Learning curves of validation loss for different activation functions in case of flower species classification

IV. Proposed Methodology

Two different methodologies have been applied for different tasks. The details of these methodologies are described in this section.

A. *Task 1*

Task 1 is a 10-class classification problem. In this case, only minmax normalization is performed as preprocessing step. After preprocessing, the RGB images are fed into a deep convolutional neural network (CNN). The structure of the network and other hyperparameters have been be chosen through different experimentations.

The proposed CNN has two parts. The first part of the model performs convolution and it contains 6 hidden stages. Each stage starts with a convolutional layer followed by batch normalization, average-pooling, and dropout layer with 0.1 probability. The batch normalization mainly speeds up the training and dropout works as a regularizer. 'ReLU' has been used as an activation function. The number of filters used in the layers are 16, 32, 64, 128, 256, and 512, respectively. The size of the filter is 4×4. Stride operations of 2×2 is used in each convolution operation.

After the convolution part, there are two hidden layers where each hidden layer is followed by batch normalization and dropout with 10% probability. The hidden layers contain 256 and 128 neurons, respectively. The network ends with a fully connected layer with 10 neurons and softmax activation function.

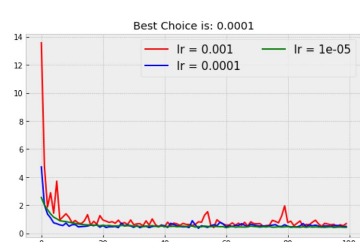

Fig. 5: Learning curves of validation loss for different learning rates in case of flower species classification

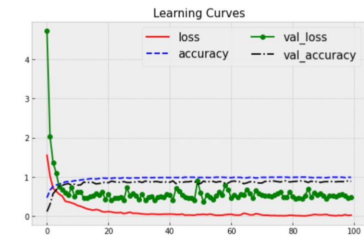

Fig. 6: Learning curves of training and validation loss and accuracy for final model in case of flower species classification

	precision	recall	f1-score	support
0.0	0.98	0.98	0.98	177
1.0	0.98	1.00	0.99	180
2.0	0.99	0.95	0.97	205
3.0	1.00	1.00	1.00	140
4.0	0.97	0.99	0.98	173
5.0	0.99	0.99	0.99	156
6.0	0.98	0.99	0.99	160
7.0	0.98	0.99	0.99	172
8.0	1.00	0.99	1.00	162
9.0	0.99	0.97	0.98	133
accuracy			0.99	1658
macro avg	0.99	0.99	0.99	1658
weighted avg	0.99	0.99	0.99	1658

Fig. 7: Classification report for train set in case of flower species classification

B. *Task 2*

The proposed methodology for this task is shown in Figure 1. This task is performed in two stages. First, the Ymax and Ymin values are predicted. Then, with the help of this prediction, Xmax and Xmin values are calculated. For Y-axis coordinates prediction, RGB image is used. The images are gone through preprocessing, which included 60% size reduction and minmax normalization. These images are fed into a CNN architecture and Y-axis coordinates are predicted. In the second stage, instead of RGB image, HOG (Histogram of Oriented Gradients) images are used. These images go through the same preprocessing steps. Then, the pixel that location outside the predicted Ymin and Ymax values are made zero. This step is called masking. The masked HOG images are fed into the second CNN and X-axis coordinates are detected. Last of all, post-processing is performed to predict the bounding box of the car.

The structure of the network and other hyperparameters have been be chosen through different experimentations. The first CNN has two parts. The first part of the model performs convolution and it contains 6 hidden stages. Each stage, except last one, starts with a convolutional layer followed by batch normalization, average-pooling, and dropout layer with 0.1 probability. The batch normalization mainly speeds up the training and dropout works as a regularizer. 'ReLU' has been used as an activation function. The number of filters used in the layers are 32, 64, 128, 256, 512, and 1028 respectively. The size of the filter is 4×4. Stride operations of 2×2 is used in each convolution operation. After the convolution part, there are three hidden layers where each hidden layer is followed by batch normalization and dropout with 10% probability. The hidden layers contain 256, 128, and 64 neurons, respectively. The network ends with a fully connected layer with 4 neurons and 'Sigmoid' activation function. The scond neural network resembles the first one.

V. Experiment and Results

A. *Training Scheme*

In the training phase, Categorical cross-entropy and mse are used as loss function for task 1 and task, respectively. Mini batch optimization has been used. Adam stochastic

	precision	recall	f1-score	support
0.0	0.89	0.88	0.88	48
1.0	0.95	0.89	0.92	44
2.0	0.76	0.70	0.73	46
3.0	0.92	0.97	0.95	36
4.0	0.81	0.87	0.84	45
5.0	0.88	0.90	0.89	40
6.0	0.84	0.95	0.89	43
7.0	0.83	0.95	0.89	37
8.0	0.91	0.91	0.91	32
9.0	0.97	0.77	0.86	44
accuracy			0.87	415
macro avg	0.88	0.88	0.87	415
weighted avg	0.88	0.87	0.87	415

Fig. 8: Classification report for test set in case of flower species classification

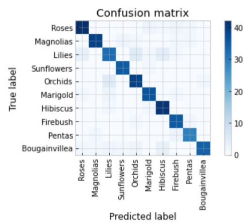

Fig. 9: Confusion matrix for test set in case of flower species classification

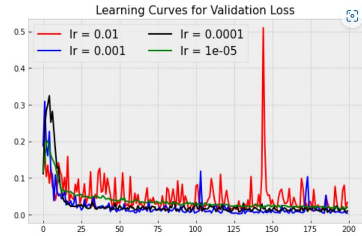

Fig. 12: Learning curves for validation loss for Ymax and Ymin prediction of car

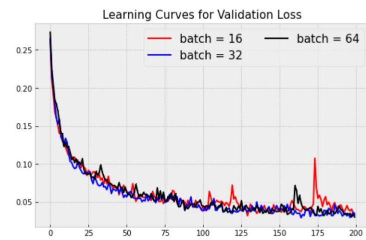

Fig. 13: Learning curves for validation loss for Ymax and Ymin prediction of car

optimization algorithm [1] is used. The learning rate and batch sizes are decided through experiments.

The performances of classification tasks are decided in terms of accuracy, precision, recall, and F_1-score. Confusion matrix is used for visualization. On the other hand, mse value is calculated for car bounding box detection. As a qualitative measure, intersection over unio (IOU) score is claculated.

B. Results Analysis

1) Task 1: In order to find the best architecture and hyper-parameter, the validation learning curves have been observed. Figure 2 shows the learning curves of validation loss for different number of hidden layers. As we can see from the figure, the validation losses for 6 hidden layers have been lesser than other architectures. Thus, the best choice of number of hidden layer is 6. Next, the filter size have been changed from 3 to 5. The corresponding result is plotted in Figure . From the figure, it can be noted that the best choice of filter size for this classification task is 4.

The learning curves of validation loss for different activation functions used in hidden layers are shown in Figure 4. Though the performances are seemed to be very close, the 'ReLU' performs best. Moreover, Figure 5 shows the learning curves of validation loss for different learning rates. From the figure, it can be seen that the best choice of learning rate is 0.001. The learning curves of training and validation loss and accuracy for final model in the case of flower species classification are shown in Figure 6.

The classification reports on training and test sets are shown in Figure 7 and Figure 8. From the report, the test accuracy is found to be around 87%. In the test set, only 'Lilies' have an

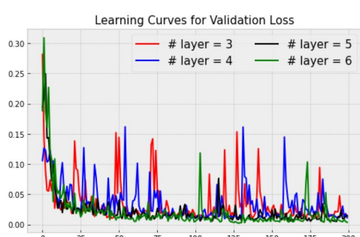

Fig. 10: Learning curves for validation loss for Ymax and Ymin prediction of car

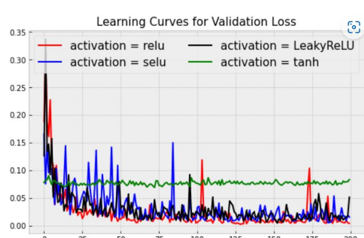

Fig. 11: Learning curves for validation loss for Ymax and Ymin prediction of car

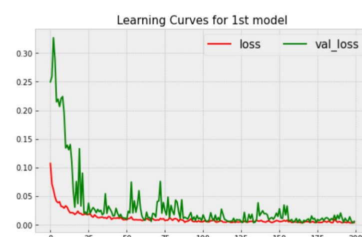

Fig. 14: Learning curves for final model for Ymax and Ymin prediction of car

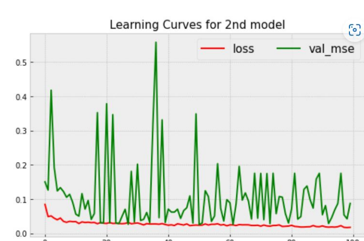

Fig. 15: Learning curves for final model for Xmax and Xmin prediction of car

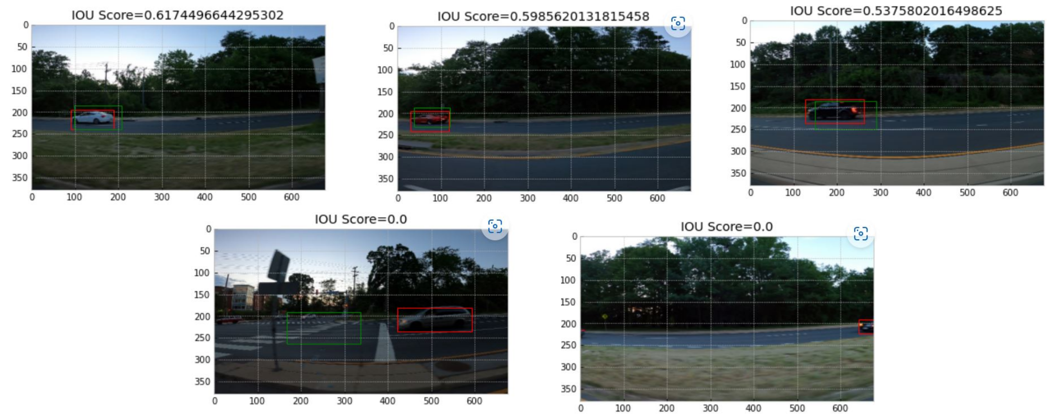

Fig. 16: Performance analysis of car detection

TABLE I: Performance analysis of car detection task

Metrics	MSE	IOU
Validation	0.0259	0.448
Test	0.0407	0.547

	precision	recall	f1-score	support
Car Absent	0.81	0.93	0.86	89
Car Present	0.90	0.73	0.81	75
accuracy			0.84	164
macro avg	0.85	0.83	0.84	164
weighted avg	0.85	0.84	0.84	164

Fig. 17: Classification report for car detection analysis

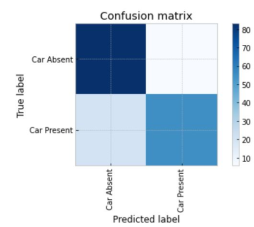

Fig. 18: Confusion matrix for car detection analysis

accuracy of less than 80%. Because, the flowers in the images of this class are of different colors, shapes, and sizes. The confusion matrix for test set is shown in Figure 9.

2) Task 2: In order to find the best architecture and hyperparameter, the validation learning curves for the 1st model have been observed. Figure 10 shows the learning curves of validation loss for predicting Ymax and Ymin with different number of hidden layers. As we can see from the figure, the validation losses for 5 and 6 hidden layers are comparable. But the losses for 6 hidden layers have been lesser. Thus, the best choice of number of hidden layers is 6. The learning curves of validation loss for predicting Ymax and Ymin with different activation functions used in hidden layers are shown in Figure 11. Though the performances are seemed to be very close except 'tangent', the 'ReLU' activation function performs best. Moreover, the learning curves of validation loss for predicting Ymax and Ymin with different learning rates and batch sizes are shown in Figure 12 and Figure 13, respectively. From the figures, it can be seen that the best choices of learning rate and batch size are 0.001 and 32, respectively.

The learning curves of training and validation losses of final models for predicting Ymax and Ymin are shown in Figure 14, while the learning curves for predicting Xmin and Xmax are shown in Figure 15. It can be observed that the losses for Ymax and Ymin prediction are lesser in comparison to prediction of Xmax and Xmin.

The performance of the proposed methodology is shown in Table 1 and Figure 16. It shows 5 instances of predicted and ground truth bounding boxes of car. The ground truth and prediction are marked in red and green color, respectively. The IOU scores are mentioned at the top of each of the figures. It can be seen that for all the 3 images in the first row, the predicted bounding box is pretty close. In the second row, the prediction has failed. In the first image of the second row, there are multiple cars in the figures. But since I have considered only one car in the annotation, the IOU score has been found 0 though the predicted bounding box also contains a smaller car. In the last image, a portion of the car can be noticed. In such cases, the proposed methodology has completely failed by detecting no car in the image.

If we formulate the problem as a classification problem, where the presence of car has to be tracked, the calculated results of the proposed method are shown in Figure 17 and Figure 18. The figure shows that in almost 84% cases the proposed method is correct in detecting the presence of car in the image.

VI. Observations and Conclusion

Classification and object detection are two important tasks in machine learning. In this task, these problems are explored for two important cases: flower species classification and car detection. From the experiments following observations can be made:

- Only 1000 images are not good enough to train network for car bounding box detection. Data augmentation can play a vital role here.
- The proposed method can detect the no-car images very effectively but in multiple car instances, it failed. Because, multiple car annotations are not covered in training set.
- IOU used in this work is a good metric to predict the overlap between the true and predicted bounding box. Average precision can be another good option.
- In case of no label test car detection dataset, semi-supervised methods can be used.
- For flower species classification, the observed performance is good but it can be improved using data augmentation.
- Transfer learning is not performed in this work. In future, that should be expolored.

References

[1] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learning Representations*, Banff, Canada, 2014.