Transportation Vehicle Type Classification

1st Ali El-Sherif
School of Engineering
University of Ottawa
aelsh054@uottawa.ca

2nd Amjad Dife
School of Engineering
University of Ottawa
adife035@uottawa.ca

3rd Basma Abd-Elwahab
School of Engineering
University of Ottawa
babde014@uottawa.ca

4th Shrief Ahmed
School of Engineering
University of Ottawa
sahme277@uottawa.ca

***Abstract*—Transportation is one of the most essential life aspects. Machine learning has added a lot through classification, detection, and segmentation. Multiple deep convolutional neural networks have been used to classify 10 images of transpiration vehicle types, which have been collected from the internet. There are 50 images for each class, so the total size of the data is 500 images. The data has been split into training, validation and testing. The training size is 80% (including 20% validation) and 20% for testing. Three models have been used for the experiments. The baseline CNN model, VGG-16 model with freezing all the weights of the network and adding new layers, and a tuned VGG-16 model with the last 2 layers unfreezed to be used in fine-tuning to measure the performance of the models with the original data, the effect of data augmentation, the effect of L1 and L2 regularization, and the effect of data augmentation and regularization on the models. To enhance the models' accuracies, ensemble method has been applied four times, the first experiment combined the three original models using the original data, second experiment combined the three original models using the augmented data, the third experiment combined the original data with L1 and L2 regularization to prevent overfitting. The last experiment combined the augmented data with L1 and L2 regularization. The results showed that the best model in validation is the ensemble method using L1 and L2 regularization with data augmentation, and in testing, the ensemble method using the original dataset without modifications.**

I. Introduction

The traditional machine learning techniques are not enough to deal with images and achieve high accuracy. Computer vision has added a lot through classification, detection, and segmentation. Transportation is one of the most essential life aspects. This paper deals with images of different types of transportation vehicles as its data and classify the input picture which can be useful in many industries like surveillance cameras, traffic systems, auto-driven car systems and many more. Vehicle production factories are increasing every year. It is becoming increasingly difficult to deal with all vehicle types that are driven on public roads and track them solely using human abilities. With the help of new computer technologies like deep neural networks and computer vision that enable the machine to see like humans, keeping track of all that vehicles and keep the citizens safe as safe as possible from illegal or unauthorized vehicles. Three models have been applied to classify the vehicle types. The first model is the convolutional neural network (CNN) model, which is used for processing data that has an input shape like images. This model has been initialized with12 layers to classify the images data. The VGG-16 model which consists of 16 layers, has the ability to perform image classification with the principle of transfer learning. Freezing some layers or weights can be applied to increase the model performance. Data augmentation has been applied to increase the images size, as the images size is too small. Also, regularization has been applied to get the best performance that could be used in a real-time system. Ensemble method combines the predictions of the models to improve the performance of the models, which is more robust than the individual models, it tries to capture complementary information from the three models and reduce the error of classifying the ten classes. The performance of the ensemble method is the best one, which achieved a high accuracy is training and validating the model.

II. Related Work

A. Implementation of CNN for Plant Leaf Classification [1]

Problem Definition: Image classifcation of medicinal plants using convolutional neural network.

Introduction and claims: The authors in this study aimed to detect medicinal plants by building a classification model for medical experts or even normal people which can be used through mobile phone cameras. this study implemented an image classification with a deep learning model for planet leaves which people can use to determine the different types of medicinal plants. The study has focused on 5 types of medical plants to be recognized.

Method: The authors have used CNN architecture. In the training process, three layers were used: The convolutional layer, the Pooling layer and the Fully connected layer. In the convolutional layer, all data features are reduced to get the most important features in every layer. The convolutional layer complexity was then optimized through three parameters depth, stride and zero padding. In the pooling layer, they used average pooling and max pooling with 2x2 size and stride equal to 2. In the fully connected layer, they added several hidden layers like the activation function, loss function and an output layer. Finally, dropout was used to prevent overfitting.

Results: The data has been tested on 500 images "100 for every class". The training and validation stages were used through 28x28 pixel images with 5 and 7 layers. That resulted an accuracy of 86%, F1-score of 23%, precision of 22% and recall of 24%. They have found that the accuracy increases with adding more hidden layers.

B. A simple and effective method for image classification [2]

Problem: Image classification of handwriting using deep neural networks and pre-trained models.

Introduction: CNN is a feed forward neural network that consists of convolutional layers, pooling layers and full connecting layers. The deep neural networks (DNNs) specifically the trained model LetNet-5 recognized the hand-written well, moreover the AlexNet model has a deeper and wider architecture than LetNet-5 architecture. There are many applications of models, such as VGGNet, which is more complex than AlexNet, Network In Network (NIN) which is based on the multilayer perceptron model, Google Inception Net, which contains 22 layers deep network, and ResNet model which contains 152 layers.

Methodology: The authors used a complicated convolutional neural network (CNN) for image classification problem, the model consists of two combined models, CNN model and the ELM model. The CNN model used for feature extraction and the ELM model used to classify the images with the softmax activation function. The CNN model consists of 2 convolutional layers, the first has 6 nodes, and the second has 12 nodes. The ELM model assigned the parameters randomly. This model is efficient in small and medium sized images.

Results: This experiment applied on cifar-10 dataset, used 50000 images for training and 10000 images for testing and compared with another 4 datasets, using MATLAB, the accuracy increased and the running time decreased, this way is efficient in image classification and easy to implement.

C. Image Classification using Convolutional Neural Networks [3]

Problem: Image classification using convolutional neural network.

Introduction and claims: The authors defined image classification as a task of extracting information from images and claimed that it could be achieved by labeling the pixels of the images in different classes, nevertheless, CNNs do not require the features to be hand-engineered. It was mentioned that many research works have been done to give the computer the same capability as humans for understanding features from images, but they focused only on the low-level feature of the images.

Methods: The MNIST dataset was used which consists of 28x28 gray-scaled images, the first layer in their system architecture comprised 32 filters each of size 3x3 to be applied on the images and produce 32 feature maps of size 26x26, in the second layer they applied 64 filters each filter of size 3x3 and produce 64 feature maps of size 24x24. The third layer is a Max pooling layer which was used to down sample the images to 12x12 by using a 2x2 subsampling window. in layer 4 they built a fully connected layer with 128 neurons and a sigmoid function. RELU activation function was used with 128 Batch size and 5 epochs for training.

Results: As a result of their work, they have reached 98.42% in terms of Accuracy.

III. DATA

- The collected images dataset consists of 10 classes, (Airplane, Car, Bus, Boat, Bike, Motorcycle, Scooter, toktok, Tram and Truck). Every class consists of 50 images, so the total size of the data is 500 images.
- All the images were resized to 64*64 pixels.

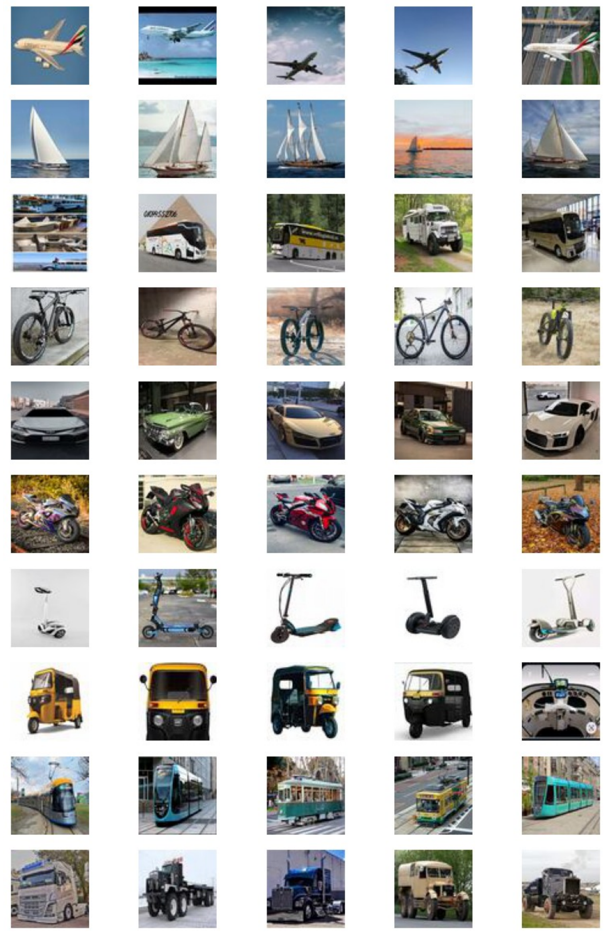

Fig. 1. Data sample

- The data has been split into 80% training (20% for validation), and 20% for testing, so every class has 30 images for training, 10 images for validation, and 10 images for testing.
- The images classes names have been replaced with numbers from 0 to 1.
- The scaling has been applied to resize the images and scaling them to be in a float type, dividing all pixel's values by the largest pixel value which is 255 to normalize them to be in range from 0 to 1.
- Because of the small size of images, data augmentation has been applied to training data and used in models, for the experiment of measuring the performance of models with data augmentation.

- Data generator parameters:
  - featurewise_center: it's a Boolean value which has been set to True, to set the input mean of the data to 0.
  - featurewise_std_normalization: it's a Boolean value which has been set to True, to divide the input images by the dataset's standard deviation.
  - rotation_range: it's an integer value of the degree range for random rotations of the input images, and has been set to 20.
  - width_shift_range: it's a float value of the random horizontal shifts of the input images, and has been set to 0.2.
  - height_shift_range: it's a float value of the random vertical shifts of the input images, and has been set to 0.2.
  - horizontal_flip: it's a Boolean value, and has been set to True for randomly flip input images horizontally.
  - validation_split: it's a float value which must be between 0 to 1 to reserve the validation images set.

IV. Methods

To perform image classification on the "Transportation vehicles" dataset, a three-model based experiment has been made to study the effect of different regularization techniques and data augmentation on the accuracy of each classifier in fig(2). Three models were used, firstly, in fig (3),a sequential CNN model with four convolution layers the first two convolution layers with 64 filters of size (3,3) and the third and last convolution layer with 128 filters of the same size, a ReLU activation function has been used in each convolution layer to add the nonlinearity feature to the classifier and increase its complexity, each convolution layer followed by a (2,2) max-pooling layer to reduce the size of the feature maps resulting from applying the filters on each convolution layer to the image[4]. To extract the high-level features and perform classification, a fully connect layer with 512 neurons with dropout equal to 0.5 and ReLU activation function have been used, finally to perform multi-class classification a fully connected layer with 10 neurons and SoftMax activation function has been used. Secondly, in fig (4), a classifier was built using the layers of the VGG16 network as a feature extractor by freezing all the weights so it cannot be updated during the training phase, nevertheless, transfer learning idea has been adapted to both of second and third classifiers. Thirdly, in fig (5), a classifier was built the layers of VGG16 also as a feature extractor but with the weights in the last two layers unfrozen. Custom classification layers have been added to the top of VGG16 layers so they can adapt to the current classification problem. Four Regularization techniques have been used to study the effect of each of them on each classifier's evaluation metrics. Firstly, by using data augmentation techniques to handle the problem of the small size of the training and validation data. Secondly, by adding regularization penalties on one of the fully connected layers in each model's architecture. Thirdly, by implementing the Ensemble method in which the predictions of each classifier are collected and used to find the classification based on the most appeared class label in the result. The Ensemble method has been used to reduce the error of each classifier since each classifier has different misclassifications than the others[5].

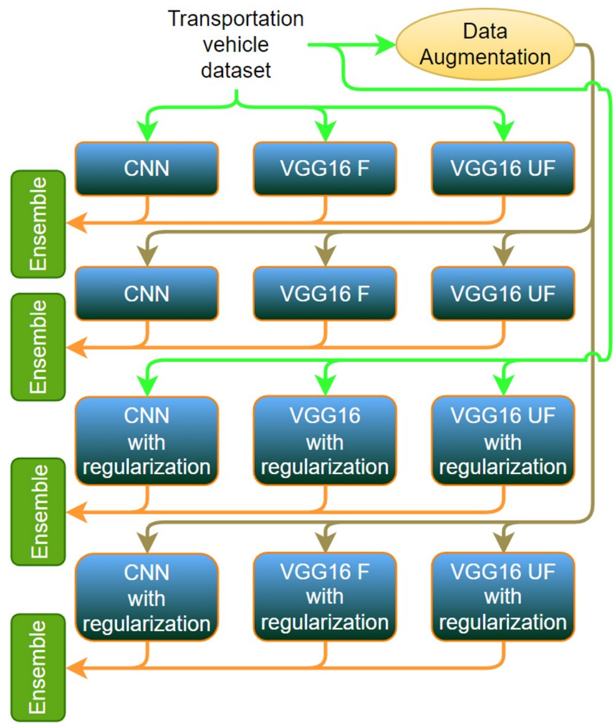

Fig. 2. System architecture

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 64)        1792

 max_pooling2d (MaxPooling2D  (None, 31, 31, 64)       0
 )

 conv2d_1 (Conv2D)           (None, 29, 29, 64)        36928

 max_pooling2d_1 (MaxPooling  (None, 14, 14, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 12, 12, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 6, 6, 128)        0
 2D)

 conv2d_3 (Conv2D)           (None, 4, 4, 128)         147584

 max_pooling2d_3 (MaxPooling  (None, 2, 2, 128)        0
 2D)

 flatten (Flatten)           (None, 512)               0

 dropout (Dropout)           (None, 512)               0

 dense (Dense)               (None, 512)               262656

 dense_1 (Dense)             (None, 10)                5130

=================================================================
Total params: 527,946
Trainable params: 527,946
Non-trainable params: 0
```

Fig. 3. CNN architecture

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 64, 64, 3)]       0

 block1_conv1 (Conv2D)       (None, 64, 64, 64)        1792

 block1_conv2 (Conv2D)       (None, 64, 64, 64)        36928

 block1_pool (MaxPooling2D)  (None, 32, 32, 64)        0

 block2_conv1 (Conv2D)       (None, 32, 32, 128)       73856

 block2_conv2 (Conv2D)       (None, 32, 32, 128)       147584

 block2_pool (MaxPooling2D)  (None, 16, 16, 128)       0

 block3_conv1 (Conv2D)       (None, 16, 16, 256)       295168

 block3_conv2 (Conv2D)       (None, 16, 16, 256)       590080

 block3_conv3 (Conv2D)       (None, 16, 16, 256)       590080

 block3_pool (MaxPooling2D)  (None, 8, 8, 256)         0

 block4_conv1 (Conv2D)       (None, 8, 8, 512)         1180160

 block4_conv2 (Conv2D)       (None, 8, 8, 512)         2359808

 block4_conv3 (Conv2D)       (None, 8, 8, 512)         2359808

 block4_pool (MaxPooling2D)  (None, 4, 4, 512)         0

 block5_conv1 (Conv2D)       (None, 4, 4, 512)         2359808

 block5_conv2 (Conv2D)       (None, 4, 4, 512)         2359808

 block5_conv3 (Conv2D)       (None, 4, 4, 512)         2359808

 block5_pool (MaxPooling2D)  (None, 2, 2, 512)         0

 flatten (Flatten)           (None, 2048)              0

 dense_2 (Dense)             (None, 400)               819600

 dense_3 (Dense)             (None, 300)               120300

 dropout_1 (Dropout)         (None, 300)               0

 dense_4 (Dense)             (None, 10)                3010

=================================================================
Total params: 15,657,598
Trainable params: 942,910
Non-trainable params: 14,714,688
_________________________________________________________________
```

Fig. 4. Freezed VGG16 architecture

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 64, 64, 3)]       0

 block1_conv1 (Conv2D)       (None, 64, 64, 64)        1792

 block1_conv2 (Conv2D)       (None, 64, 64, 64)        36928

 block1_pool (MaxPooling2D)  (None, 32, 32, 64)        0

 block2_conv1 (Conv2D)       (None, 32, 32, 128)       73856

 block2_conv2 (Conv2D)       (None, 32, 32, 128)       147584

 block2_pool (MaxPooling2D)  (None, 16, 16, 128)       0

 block3_conv1 (Conv2D)       (None, 16, 16, 256)       295168

 block3_conv2 (Conv2D)       (None, 16, 16, 256)       590080

 block3_conv3 (Conv2D)       (None, 16, 16, 256)       590080

 block3_pool (MaxPooling2D)  (None, 8, 8, 256)         0

 block4_conv1 (Conv2D)       (None, 8, 8, 512)         1180160

 block4_conv2 (Conv2D)       (None, 8, 8, 512)         2359808

 block4_conv3 (Conv2D)       (None, 8, 8, 512)         2359808

 block4_pool (MaxPooling2D)  (None, 4, 4, 512)         0

 block5_conv1 (Conv2D)       (None, 4, 4, 512)         2359808

 block5_conv2 (Conv2D)       (None, 4, 4, 512)         2359808

 block5_conv3 (Conv2D)       (None, 4, 4, 512)         2359808

 block5_pool (MaxPooling2D)  (None, 2, 2, 512)         0

 flatten (Flatten)           (None, 2048)              0

 dense_5 (Dense)             (None, 400)               819600

 dense_6 (Dense)             (None, 300)               120300

 dropout_2 (Dropout)         (None, 300)               0

 dense_7 (Dense)             (None, 10)                3010

=================================================================
Total params: 15,657,598
Trainable params: 3,302,718
Non-trainable params: 12,354,880
```

Fig. 5. Unfreezed VGG16 architecture

V. EXPERIMENTS

A four staged experiment has been conducted to perform Transportation vehicle type classification. The main objective of the entire experiment is to improve the accuracy of the classifier. In the first part of the experiment, the original training dataset was used to train each classifier of the three classifiers. Figure (6) shows the accuracy and loss curves for CNN classifier using the original data, at the earlier training epochs, both the training and validation accuracy were increasing until epoch 30, the training accuracy kept increasing while the validation accuracy decreased. In terms of the loss, Both the training and validation loss were decreasing until epoch 30, the training loss kept decreasing while the validation loss increased. Figure (7) shows the accuracy and loss curves for the second classifier (VGG16 with all the layers frozen), as the figure shows the model overfits earlier than the CNN model since both the training and validation loss were decreasing until epoch 20, the training loss kept decreasing while the validation loss increased. The value of the validation accuracy is higher than the validation accuracy of the CNN classifier. Figure (8) shows the accuracy and loss curves for the third classifier (VGG16 with the last two layers unfrozen) using the original data, as the figure shows the model overfits earlier than both the CNN model and model 2 (frozen VGG16), *since both the training and validation loss were decreasing until epoch 3 the training loss kept decreasing while the validation loss increased. The value of the validation accuracy is higher than the validation accuracy of the CNN classifier but lower than the validation accuracy of the frozen VGG16 model. In the last step in part 1 of the experiment, an ensemble learning technique has been implemented since the test dataset has been used to get the prediction of each classifier then, the prediction of each classifier has been collected and used to find the most appearance class for each instance in the test dataset.

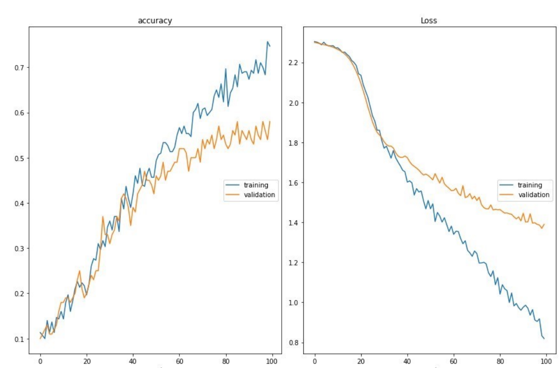

Fig. 6. Accuracy and loss for CNN using the original data(training and validation)

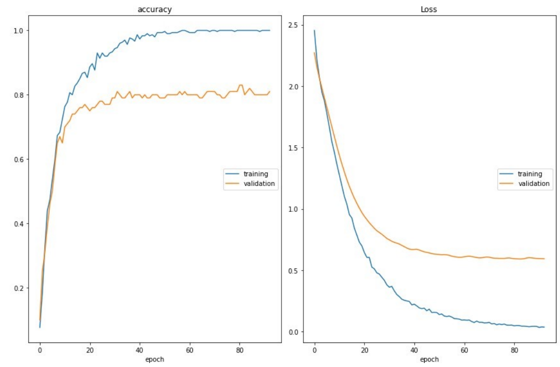

Fig. 7. Accuracy and loss for freezed VGG16 using the original data(training and validation)

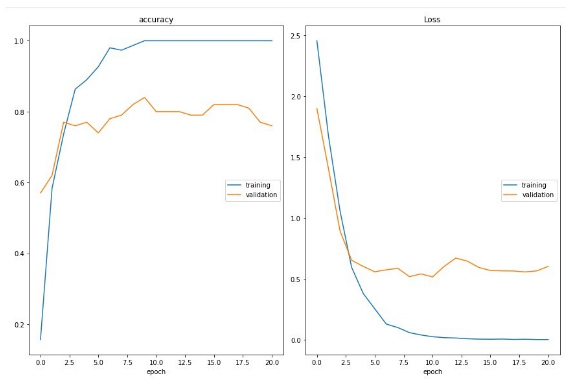

Fig. 8. Accuracy and loss for unfreezed VGG16 using the original data(training and validation)

In the second part of the experiment, the data augmentation operations have been performed to create new images for both the training and validation. The new augmented data has been created “on the fly” since it had not been saved on the disk but just it kept in the application memory. Figure (9) shows the accuracy and loss curves for the CNN classifier using the augmented data, as the figure shows the values of the loss for both training and validation got higher than the values of loss using the original data, on the other hand, the difference between the training and validation loss got lower than the difference of losses using the original data. Unlike the training accuracy with the original data, the training accuracy of the CNN classifier with the augmented data did not reach 100%, and the difference between the accuracies is not as high as the difference between the accuracies and the original data. In terms of the loss, Both the training and validation loss were decreasing until epoch 30, the training loss kept decreasing while the validation loss increased. Figure (10) shows the accuracy and loss curves for the second classifier (VGG16 with all the layers freezed) using the augmented data, as the figure shows training accuracy of the model using the augmented data got lower than the model with the original data since the model had 100% training accuracy with the original data. The model was still overfits since both the training and validation loss were decreasing until epoch 15, the training loss kept decreasing while the validation loss increased. Figure (11) shows the accuracy and loss curves for the third classifier (VGG16 with the last two layers unfreezed) using the augmented data, as the figure shows the model still overfits, but on the other hand, the value of the training accuracy of the model got lower than its value using the original data, also the validation accuracy increased since the validation accuracies of the model were 76%, 79% with the original, augmented data, respectively. In the last step in part 2 of the experiment, an ensemble learning technique was implemented the same way as last step in part 1.

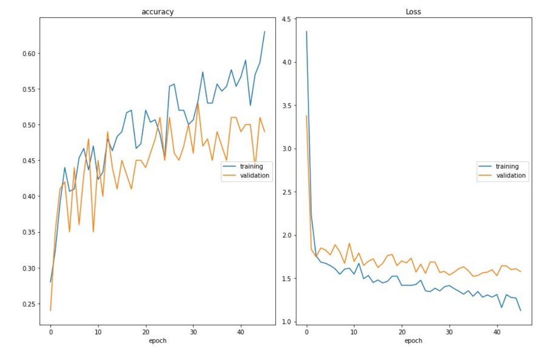

Fig. 9. Accuracy and loss for CNN using the augmented data

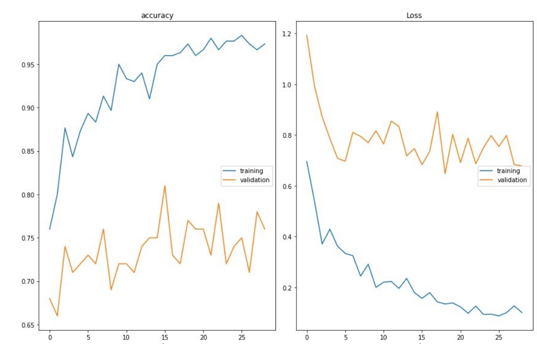

Fig. 10. Accuracy and loss for freezed VGG16 using the augmented data

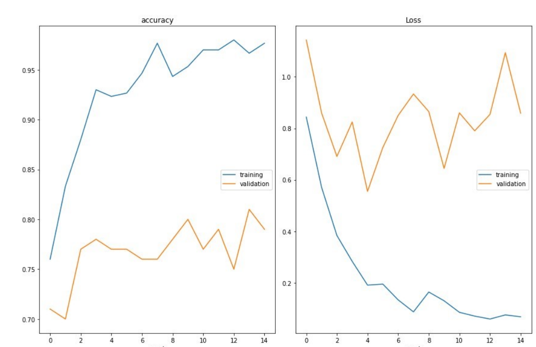

Fig. 11. Accuracy and loss for unfreezed VGG16 using the augmented data

In the third part of the experiment, L1, L2 penalties have been added using Keras library to one of the fully connected layers in each classifier of the three classifiers. Then the original data set has been used to train the model after adding the regularization terms. Figure (12) shows the accuracy and loss curves for the CNN classifier using the original data, after adding L1 and L2 regularization terms, as the figure shows the difference between the training and validation loss decreased but still the values of the loss for both training and validation is high. The model no longer overfits since it completed all the 100 epochs, as the figure shows both the training and validation loss kept decreasing together during the training phase. Figure (13) shows the accuracy and loss curves for the second classifier (VGG16 with all the layers frozen) using the original data, after adding L1 and L2 regularization. As the figure shows the model no longer overfits since it completed all the 100 epochs, and both the training and validation loss kept decreasing together during the training phase. Figure (14) shows the accuracy and loss curves for the third classifier (VGG16 with the last two layers unfrozen) using the original data after adding L1 and L2 regularization terms. As the figure shows, the model no longer overfits, the validation accuracy got higher than the other models in the previous parts. In the last step in part 3 of the experiment, an ensemble learning technique was implemented the same way as last step in part 1 and 2.

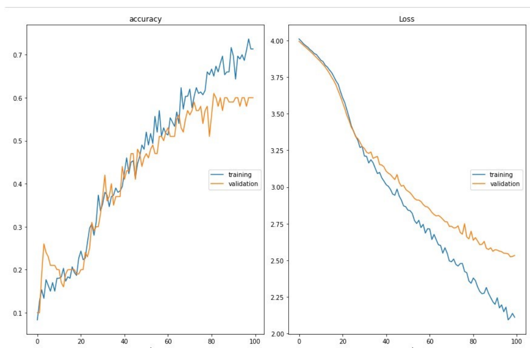

Fig. 12. Accuracy and loss for CNN using the original data after adding L1 and L2 regularization terms

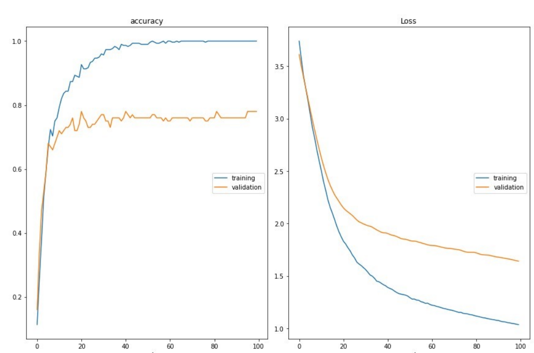

Fig. 13. Accuracy and loss for freezed VGG16 using the original data after adding L1 and L2 regularization terms

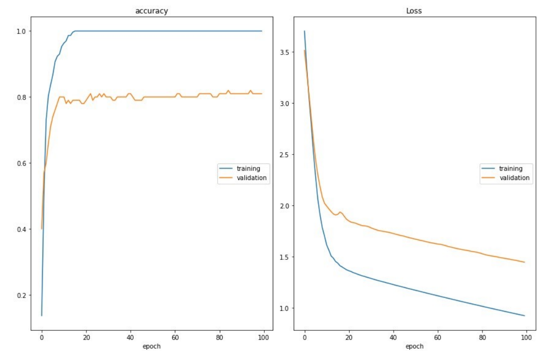

Fig. 14. Accuracy and loss for unfreezed VGG16 using the original data after adding L1 and L2 regularization terms

In the last part of the experiment, the augmented data has been used to train the models after adding L1, L2 penalties. Figure ( ) shows the accuracy and loss curves for the CNN classifier using the augmented data, after adding L1 and L2 regularization terms, as the figure shows the model overfits starting from epoch 70 since the training loss kept decreasing while the validation loss increased again. Figure ( ) shows the accuracy and loss curves for the second classifier (VGG16 with all the layers frozen) using the augmented data, after adding L1 and L2 regularization. As the figure shows the model overfits starting from epoch 20 since the training loss kept decreasing while the validation loss increased again. Figure ( ) shows the accuracy and loss curves for the third classifier (VGG16 with the last two layers unfrozen) using the augmented data after adding L1 and L2 regularization terms. As the figure shows, at epoch 30 the model started to overfit. In the last step in part 4 of the experiment, an ensemble learning technique was implemented the same way as last step in part 1, 2, and 3.

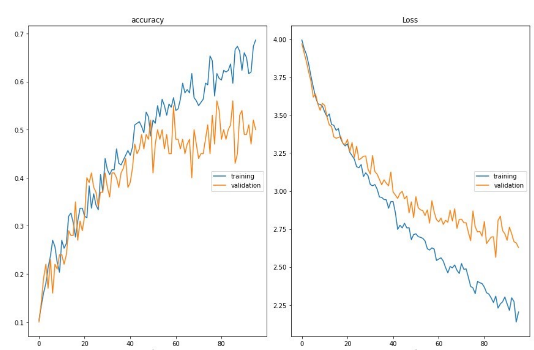

Fig. 15. Accuracy and loss for CNN using the augmented data after adding L1 and L2 regularization terms

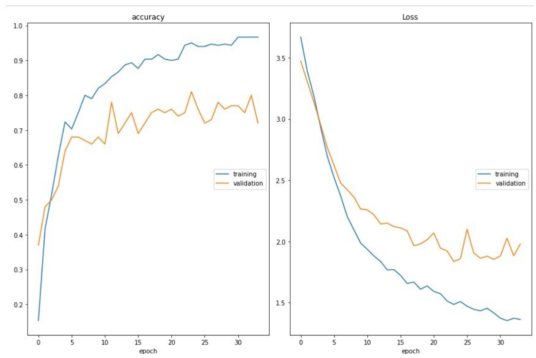

Fig. 16. Accuracy and loss for freezed VGG16 using the augmented data after adding L1 and L2 regularization terms

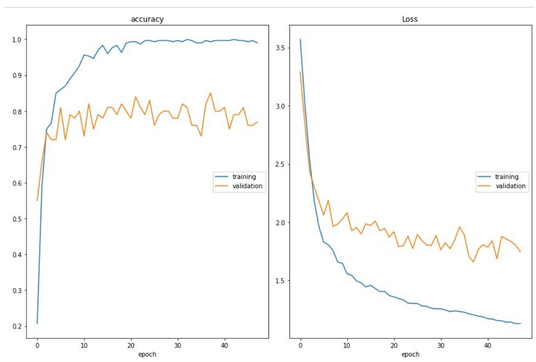

Fig. 17. Accuracy and loss for unfreezed VGG16 using the augmented data after adding L1 and L2 regularization terms

The champion model is tuned VGG16 with L1, L2 regularization and its confusion matrix is in fig(18):

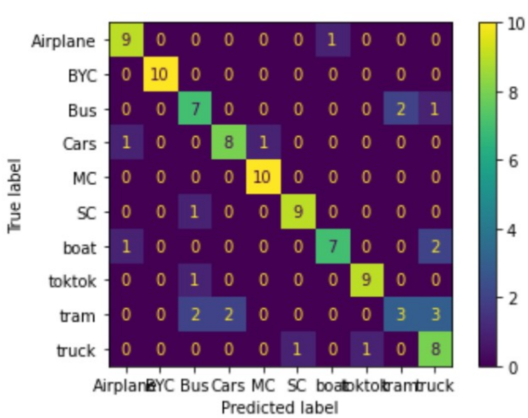

Fig. 18. Confusion matrix of champion model

The model has achieved 80% accuracy. reached in some classes 100% but failed to predict class "tram" and achieved only 40% in F1-Score. overall, the model got predictions for the rest of the classes very well.

Figure(19) show summary of the results of CNN classifier through the four parts of the experiment. As the figure shows, both the training and validation loss of the CNN model increased with regularization terms and the hybrid approach that combine the L1, L2 penalties and data augmentation. The best training and validation accuracy for the CNN model at the original data approach. Figure(20) show summary of the results of model 2 (frozen VGG16) through the four parts of the experiment. As the figure shows, both the training and validation loss of model 2 (frozen VGG16) increased with regularization terms and the hybrid approach. The best validation accuracy for model 2 at the original data approach. Figure(21) show summary of the results of model 3 (VGG16 with last two layers unfrozen) through the four parts of the experiment. As the figure shows, both the training and validation loss of model 3 (VGG16 with last two layers unfrozen) increased with regularization terms and the hybrid approach. The best validation accuracy for model 3 at the hybrid approach. The validation loss decreased at the approach that add L1, L2 regularization terms to the model.

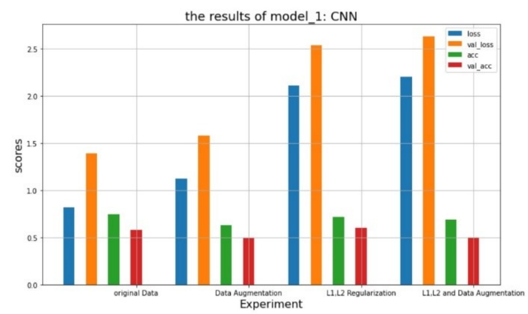

Fig. 19. CNN Evaluation Results

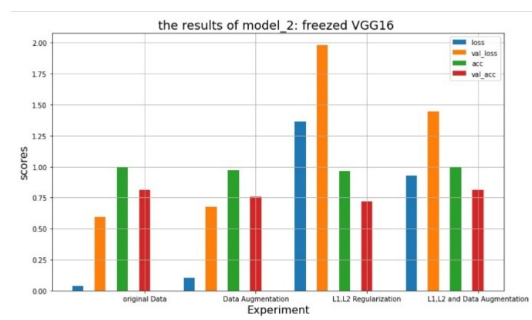

Fig. 20. Freezed VGG16 Evaluation Results

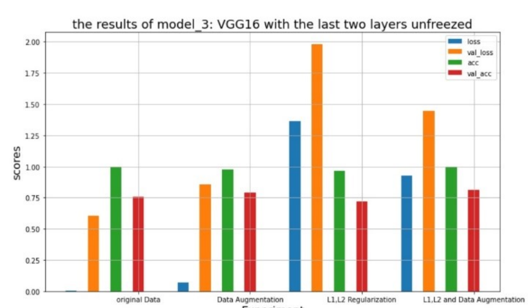

Fig. 21. unfreezed VGG16 Evaluation Results

VI. Conclusion

To perform transportation Vehicle's classification a dataset of 10 classes has been collected with 50 images with size (64*64) for each class. The data has been split into 80% training (included 20% for validation), and 20% for testing. A four-part experiment has been conducted to check the effect of four regularization techniques on three different models. Firstly, a data augmentation technique has been used to generate new instances that can be used to train and validate each model. Secondly, L1 and L2 regularization terms have been added to each model and the model trained using the original data. Thirdly, a hybrid approach that combines data augmentation and regularization has been implemented. Fourthly, after getting the prediction of each model from each part of the previous three parts an ensemble learning approach has been implemented by collecting the prediction and calculating the most appeared class for each sample in the test dataset. The use of L1 and L2 regularization terms solved the problem of overfitting and decreased the difference between the training and validation loss. Generating new data using data augmentation technique positively affected on the second and the third models (which bases on transfer learning using VGG16 network) but on the other hand, it negatively affected on the CNN architecture since it increased the value of the training and validation loss without solving the problem of overfitting. The use of ensemble method positively affected on handling the different classifier by building a strong classifier. The hybrid approach increased the value of the training and validation loss without solving the problem of overfitting. The best model is VGG16 with fine-tuning of the last two layers, with L1 and L2 regularization terms, and the model fitted on the original data, with 100% training, 81% validation, and 80% testing accuracies.

VII. Future Works

Adding more layers to the CNN model architecture to increase the model complexity. Try multiple models on the dataset such as ResNet model and EfficentNet instead of VGG16. Increase the images size using different data augmentation techniques during the training process. Deploy an application to classify the images using Streamlet or AWS Elastic Beanstalk.

Bibliography

References

[1] M. Diqi and S. H. Mulyani, "Implementation of CNN for Plant Leaf Classification," International Journal of Informatics and Computation, vol. 2, no. 2, p. 1, 2021.

[2] P. Hao, J.-H. Zhai, and S.-F. Zhang, "A simple and effective method for image classification," IEEE Xplore, Jul. 01, 2017. https://ieeexplore.ieee.org/abstract/document/8107769.

[3] M. Ramprasath, M. Vijay. Anand, and S. Hariharan, "Image Classification using Convolutional Neural Networks," International Journal of Pure and Applied Mathematics, vol. 119, no. 17, pp. 1307–1319, 2018.

[4] C.-C. Tsai, C.-K. Tseng, H.-C. Tang, and J.-I. Guo, "Vehicle Detection and Classification based on Deep Neural Network for Intelligent Transportation Applications," IEEE Xplore, Nov. 01, 2018. https://ieeexplore.ieee.org/document/8659542 (accessed Jan. 05, 2022).

[5] M. Aamir, Z. Rahman, W. Ahmed Abro, M. Tahir, and S. Mustajar Ahmed, "An Optimized Architecture of Image Classification Using Convolutional Neural Network," International Journal of Image, Graphics and Signal Processing, vol. 11, no. 10, pp. 30–39, Oct. 2019, doi: 10.5815/ijigsp.2019.10.05.

Work done

	Tasks
Ali	Train models with regularization and data augmentation
Amjad	Apply ensemble method
Basma	Train Models with regularization
Shreif	Train the baseline model

TABLE I
Tasks done by each team member

N.B: All the tasks were divided equally on the team members.