Optimize vehicle recognition performance on Convolutional Neural Networks Using common mainstream methods In Vehicle-x dataset

Richeng ZHANG

Research School of Computer Science, Australian National University
Richeng ZHANG u7094927@anu.edu.au

Abstract. Using the dataset Vehicle-x [1] introduced by Yao, Y., Zheng, L., Yang, X., Naphade, M., & Gedeon, T contains different images from different kinds of vehicles to train a convolutional neural network to do vehicle recognition task. The classification task is to classify the different kinds of vehicles (1362 different kinds of vehicles in total). In previous experiment, just using the restricted dataset from Vehicle-x dataset to build up a neural network to try to reach a high accuracy, but that model performs bad. So, in this paper, the goal is to reach a high accuracy in vehicle recognition so, a famous convolutional neural network, Alexnet [3] will be used and then try to use many common mainstream methods including adjust hyperparameters and using different data augmentation method to optimize the CNN's performance. After using the optimization methods, the CNN can reach 78.38% accuracy, which is a quite good result for the vehicle recognition task. Finally, there will be a comparison between the traditional machine learning method (Maximum Likelihood Classification and Decision Tree Classification) and CNN and the result shows that CNN's accuracy is much higher than the traditional machine learning method. These experiments show the superiority of CNN over some traditional machine learning classification methods and CNN can have excellent performance in vehicle recognition problems.

Keywords: Optimization Methods, Convolutional Neural Network Alexnet, Maximum Likelihood Classification, Decision Tree Classification

1 Introduction

Recently, vehicle recognition is an interesting research problem. In the future, vehicle recognition can be used widely in many fields such as management of transport and public security. Vehicle-x [1] dataset is a useful dataset for training a neural network model to apply vehicle recognition because the dataset is based on Unity 3D engine. There are two main advantages of this dataset. The first one is that they are 3D models which are very similar to real vehicles. The other advantage is that Using the Unity 3D engine, it is very easy and convenient to get a new data sample and its ground truth label. There is a hugely difficult problem in deep learning field about collecting enough good data and washing data. Actually, getting a large amount of data about real vehicles is hard and long-time-cost work. If researchers just using Unity 3D engine to create 3D models instead of real vehicles, the work of getting dataset will be easier.

In Vehicle-x, there are 75516 different kinds of vehicle images in total, and they are divided into 1362 different kinds. The task of this paper is to use the convolutional neural network to extract the features from the images and let the neural network learn the features and classify different kinds of vehicles. When the convolutional neural network is training, there are many factors that affect the result's accuracy. In this paper, adjusting hyperparameters and data augmentation optimization will be used. Of course, the two biggest difficulties in this regular training task are how to use data augmentation to increase the diversity of the data and generalize the model to avoid overfitting and how to adjust the hyperparameters to make the convolutional neural network have the best performance. If the accuracy of this model is very high, it shows that this model is very practical in many fields and it can be applied in practical application to help people manage and dispatch vehicles on the highway.

The main method of evaluating the model is accuracy. The accuracy can be calculated by comparison predicted classification results from the convolutional neural network and true classification results. There are some other indexes to evaluate the convolutional neural network, for example, loss. However, these indexes can just reflect some aspects of the model, so these indexes will not be the main method of evaluation. To evaluate the entire performance of the model, a comparison between the traditional machine learning classification method and the convolutional neural network is necessary. In this paper, Maximum Likelihood Classification and decision tree classification are used because they are common methods to compare the performance introduced by Milne, L. K., Gedeon, T. D., & Skidmore, A. K. [2]. There is also a common method called the threshold method in that paper. However, this is a 1362 classification problem, the threshold method needs the researchers to adjust the threshold to avoid false positives in every class manually. It is a nearly impossible method in this 1362 classification problem. Therefore, the threshold will not be used in this paper. The purpose of doing a comparison between the model in this paper and the traditional machine learning classification method is to show how excellent the convolutional neural network performs in all classification methods.

2 Methodology

For this Vehicle-x dataset, an important evaluation method is comparison method. Because the task in this paper is trying to achieve a perfect performance in convolutional neural network, comparing with other classification can help researchers know how good or how bad the convolutional neural network performs. The architecture of the convolutional neural network, Alexnet [3] and maximum likelihood classification method and decision tree classification method will be mentioned in this part. In addition, data preparation and data preprocessing are the basic step for deep learning because excellent data processing can help decrease training time and increase accuracy of model. Therefore, necessary data preparation and data preprocessing method will also be mentioned in this methodology part.

2.1 Data preparation and Data preprocessing

The data in Vehicle-x are 256*256*3 images which have R, G, B channels. There are 1362 classes and 75516 images in total. These images are divided into 3 parts, training set, validation set and test set. The training set have 45438 images and they are used for training convolutional neural network. The validation set have 14936 images and they are used for validation which contains adjusting hyperparameters and changing data augmentation method. The test set have 15142 images and they are used for testing part and evaluation of the convolutional neural network. There is a reflection relation between the name of images and their ground truth label, and their reflection relation is stored in a xml file. Because the speed of reading the xml file in python is very slow, it is very necessary to extract the reflection relation from the xml file and save them as a csv file. After that, because the images saved in train, validation and test file are not satisfied standard input format of torchvision dataset, then making folders and naming them as each class and putting the images into right class folders is necessary. To fit the specific convolutional neural network, Alexnet, all images are resized into 224*224*3. After data preparation, in each class folder, it only contains the images belong to its class. Next step is data preprocessing part. There are many methods for data preprocessing includes normalization method, shuffle method, random cropping and random flipping. In theory, all data preprocessing method can help the model increase the accuracy and generalize the model to avoid overfitting, however, the effect of data preprocessing is one of the most important research targets in this paper. Therefore, the raw data will be saved but the processed data will not be saved as a data file.

For normalization step, the z-score normalization method will be used in this paper. The purpose of z-score normalization is to let the data range become from -1 to 1 and the data is in normal distribution. The formula for each channel for each image is following.

$$normalized\ data = (data - \mu)/\sigma$$

Shuffle method is disrupting the order of data to avoid overfitting and decreasing the extra information brought by order. Random cropping and random flipping can increase the diversity of the images to generalize the convolutional neural network and avoid overfitting.

2.2 Convolutional Neural Network, Alexnet

In this paper, a famous but simple convolutional neural network Alexnet will be used. It was introduced by Krizhevsky, A., Sutskever, I., & Hinton, G. E. in 2012, which gain the champion in ImageNet competition in 2012. The architecture of Alexnet contains 5 convolution layers, 3 max-pooling layers and 3 full-connection layers. The convolution layers are used to simulate human's brain to know the features of images. Then using activation function to connect them. The usage of activation function is to let the output and input get rid of linear and help them simulate non-linear classification better. In this convolutional neural network, activation function ReLU will be chosen. The max-pooling layers are used to collect the obvious features extracted by convolution layers and reduce the dimensions. The full-connection layers are used to map the features studied by the above layers to the space and used to be classifier. The detailed architecture of the convolutional neural network, Alexnet is shown following as figure 1.

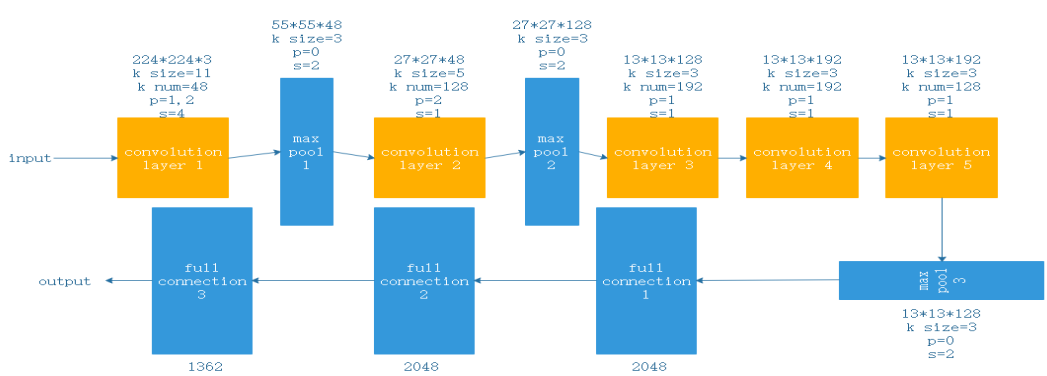

figure 1 The Alexnet's architecture

2.3 Initialization method

In convolutional neural network, initializing weight is a useful method to avoid gradient vanishing problem and gradient exploding problem. An advanced initialization method called Kaiming initialization [4] method introduced by He, K., Zhang, X., Ren, S., & Sun, J. in 2015 can improve the performance of the network a lot. Therefore, in this paper, Kaiming initialization method will be used in this convolutional neural network in order to avoid gradient vanishing problem and gradient exploding problem and make the network have better performance.

2.4 Maximum Likelihood Classification

Maximum Likelihood Classification is also called Bayesian classifier. It is a useful traditional machine learning classification method. When the dataset is big, it can summarize the different characteristic and speculate the result. For specific task in this paper, there are 1362 classification. To classify is to get $p(A|B)$ for each class. The formula is following.

$$p(A|B) = \frac{p(B|A) \times p(A)}{p(B)}$$

2.5 Decision Tree Classification

Decision tree classification is a traditional machine learning classification, and it is very similar to human doing classification. There are 1362 classification. In decision tree classification, information entropy will be used. If there are d classifications and N sample,$P_k = N_k/N$,this means that in N samples choose one sample randomly and the probability of this sample belong to kth classification. The information entropy can be shown as following.

$$Ent(D) = -\sum_{k=1}^{d} p_k \log_2 p_k$$

Then, for each divide into new class, a new information entropy can be calculated and compare with the origin information entropy to decide which characteristic can be root or leaves. For the attribution set $A = \{a_1, a_2 \dots a_N\}$, suppose D_n is the sample set which attribution equal a_n. The information gain can be shown as following.

$$Information\ gain = Ent(D) - \sum_{n=1}^{N} \frac{|D^n|}{|D|} Ent(D^n)$$

3 Results and Discussion

In this part, this paper will show the record of trying to optimize the model in the Vehicle-x dataset and change parameters to increase the accuracy. The model will be trained in the training dataset and using validation set to optimize the parameters. Finally, when all optimization methods have been used, test set can be used to evaluate how the convolutional neural network performs. In this experiment, there is an order to change parameters or data augmentation methods. If an optimization method can improve a lot, then it will be reserved. Therefore, when all optimization methods are used, the convolutional neural network can have the best performance. The origin parameters are following as a table. In every optimization method, the neural network will be trained and using validation set to how the model perform and the accuracy will be calculated an average and there are five experiments in total in each optimization method. In the previous experiment, it shows that the best choice of optimizer is Adam [5] instead of SGD, therefore, for optimizer, there will not be an experiment about it. In addition, shuffle has been proved that it can help avoid overfitting and increase accuracy, therefore, it will not be tried this time.

The Table About Initial Parameters and Data Augmentation Methods

Learning rate	Normalization	Shuffle	Epoch	Batch Size	Dropout	Random Flip/ Random Crop
0.01	False	True	30	256	0.5	False

3.1 Normalization

To normalize the training set, z-score method will be used because it can let the value between -1 and 1 and let them be in normal distribution.

The Table About Validation Accuracy vs Normalization	
Normalization	Accuracy
False	0.07%
True	0.09%

In this experiment, normalization cannot improve the performance of convolutional neural network a lot, but it still improves a little. The normalization method will be applicated in next stage of experiment.

3.2 Random Crop and Random Flip

A suitable learning rate can decide how the network perform deeply. If the learning rate is too big, it will miss the best point of result. If the learning rate is small, it will learn slowly and may be caught in local optimum. In this experiment, the different scale-value learning rate will be chosen, and the accuracy is following.

The Table About Validation Accuracy vs Random Crop and Random Flip	
Random Crop and Random Flip	Accuracy
False	0.09%
True	0.12%

The data augmentation method has better performance, therefore, in next stages of experiments, random crop and random flip will be reserved.

3.3 Learning Rate

A suitable learning rate can decide how the network perform deeply. If the learning rate is too big, it will miss the best point of result. If the learning rate is small, it will learn slowly and may be caught in local optimum. In this experiment, the different scale-value learning rate will be chosen, and the accuracy is following

The Table About Validation Accuracy vs Learning Rate	
Learning rate	Accuracy
0.01	0.12%
0.005	0.13%
0.001	0.10%
0.0005	65.36%
0.00025	67.17%
0.0001	56.33%

As the experiments' result, the learning rate between 0.0005 and 0.0001 is perfect, and the 0.00025 have the largest accuracy. Therefore, in next stages of experiments, learning rate is 0.00025.

3.4 Dropout

Dropout [6] is often used to solve overfitting problem. In the convolutional neural network, Alexnet, the dropout is set as a default value 0.5. However, in the above experiment, there is a problem about the training loss is always lager than validation loss. In this situation, it is hard to judge whether the model is in convergence because the dropout is 0.5 when it is training but the dropout is 0 when it is in validation step. In this part of experiment, what the value of dropout is suitable will be discovered.

Dropout is introduced by Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. The theory is that dropout is a parameter means probability. A neuron in neural network have probability to stop work. The probability is dropout. As empirical theory, dropout will between 0.1 and 0.5.

The Table About Validation Accuracy vs Dropout	
Dropout	Accuracy
0.1	71.17%
0.3	68.02%
0.5	67.17%

It shows that when the dropout is 0.1, it is enough to avoid overfitting, and it can achieve a high accuracy, therefore, the dropout is 0.1 in the next stage of experiment.

3.5 Batch Size

Batch size is an important hyperparameters in the convolutional neural network. Some researchers suggest that do not decay the learning rate but increase batch size can decrease the time of waiting training and get a good result [7]. However, other researchers disagree with them. They say that not everyone has the enough GPU resource and researchers should choose suitable batch size which fit their GPU resource [8]. Therefore, in this step, different batch size will be tested and find the most suitable batch size in this convolutional neural network.

The Table About Validation Accuracy vs Batch Size	
Batch Size	Accuracy
256	71.17%
128	72.73%
64	68.21%
32	65.61%

Because of limitation of GPU resource, if the batch size cannot be larger than 256. The range of batch size is from 32 to 256. The experiment shows that if the batch size is 128, the performance of the convolutional neural network is the best. So, the batch size 128 will be used in the next step of experiment.

3.6 Number of Epochs

Number of epochs represents the times of learning. If the number of epochs is small, the network will be caught in underfitting. If the number of epochs is too high, the network is often in overfitting. Therefore, the suitable number of epochs is important. The different number of epochs and their accuracy is following.

The Table About Validation Accuracy vs Number of Epoch	
Number of Epochs	Accuracy
30	71.42%
40	74.42%
50	76.99%
60	77.53%
70	78.68%
80	78.81%
90	78.94%
100	79.09%

From the table and result, when the number of epochs is 100, the accuracy is the highest obviously. However, when the number of epochs reach 70, the validation accuracy is 78.68%. Then the validation accuracy increases very slowly from this time point. Therefore, it can show that when the number of epochs reach 70, the convolutional neural network is convergent. For saving time in training part, just choose 70 as the number of epochs is suitable.

3.7 The Best Optimization Parameters in Convolutional Neural Network

According to the above experiments, the best parameters are chosen is following.

The Best Optimization Parameters in Alexnet in this Task	
Learning rate	0.00025
Normalization	True
Random Crop and Random Flip	True
Epoch	70
Shuffle	True
Dropout	0.1

3.8 The Result of Maximum Likelihood Classification and Decision Tree Classification.

For traditional machine learning, such as Maximum Likelihood Classification and Decision Tree Classification, the result shows that they perform worse than convolutional neural network. The accuracy of Maximum Likelihood Classification is just 6%. For Decision Tree Classification, the result is worse, it is just 0.42% but it cost almost 15 hours to fit the model and get the result. Therefore, for this 1362 classifications problem, decision tree classification is not practical because of the too long training time.

3.9 Evaluation and Discussion

Using the best optimization parameters above in the convolutional neural network to train and using test set to evaluate the performance. Finally, the performance of the convolutional neural network is quite good. The test accuracy is 78.38%. For a 1362 classification problem, the accuracy is ok. There are two graphs about the training loss, validation loss and validation accuracy.

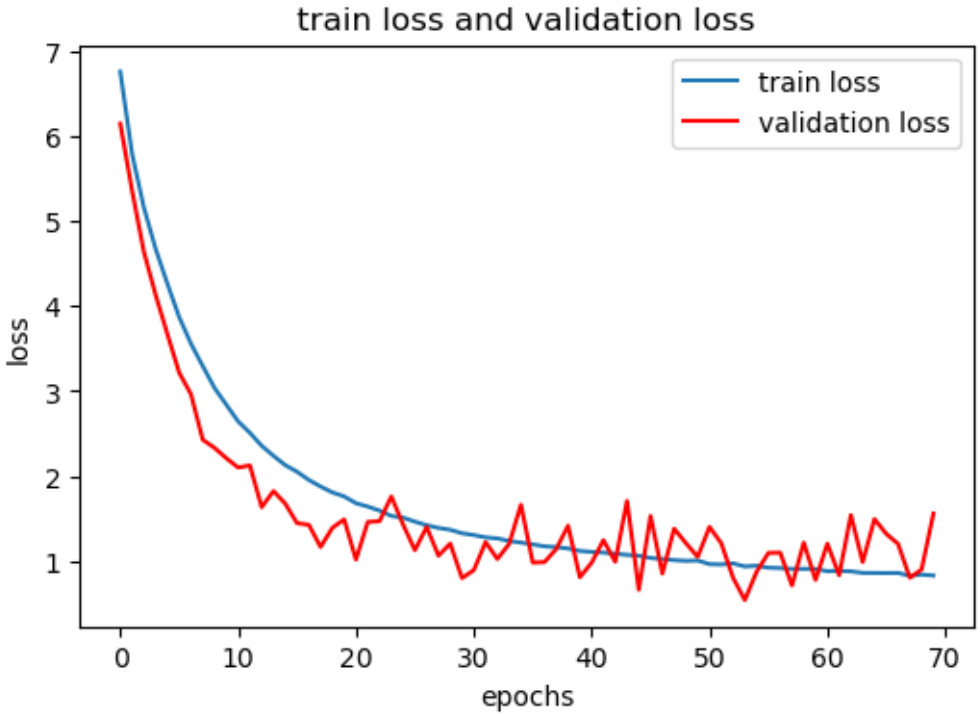

figure 2 the epochs vs train loss and validation loss

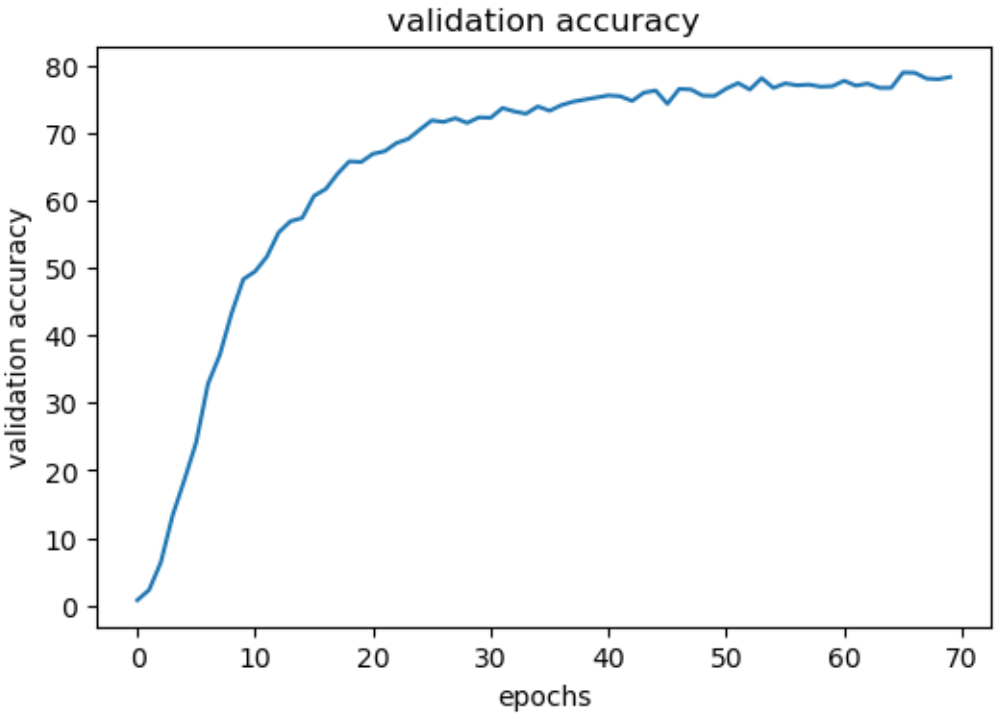

figure 3 the epochs vs validation accuracy

From the two graphs above, figure 2 and figure 3, when the epoch is growing up to 70, the validation loss is beginning to over the training loss. If the epoch continues growing, the validation loss will be growing over the training loss all the time after that point. Therefore, if the epoch is over 70, then there will be an overfitting in the model. In the graph about epoch vs validation accuracy, when the epoch is between 50 and 70, the validation of accuracy increases very slowly, and the tendency line of validation accuracy is almost flat. It shows that the model has already been in

convergence in this section. In order to eliminate error and accidental factors, the test accuracy should be between 77% and 79%.

Comparing this model with the two traditional machine learning method, Maximum Likelihood Classification and Decision Tree Classification, the performance of this model is excellent because the accuracy of these two methods is under 10% but the accuracy of convolutional neural network is almost 70% higher than these two methods.

There are two valuable problems in these experiments. The first one is a counterintuitive phenomenon about loss. The validation loss is always less than training loss in most period. In theory, validation loss should equal or larger than training loss. The second problem is that though the accuracy of this 1362 classification problem can up to 78%, it still worse than state-of-the-art level. So why the result in this paper cannot reach state-of-the-art level?

For the first problem, there are many explanations about it. The first explanation is the calculating method of different losses. For training loss, it is gotten from all loss from every batch, and it is an average of them. However, the validation loss is calculated when this epoch is over. Actually, this loss is from trained network, but the training loss is an average and it contains a lot of loss from not trained network. In this situation, training loss will be lager than validation loss. Then second explanation is that the data provided for training is too hard. There are many data augmentation methods implemented in training part such as random cropping and random flipping but for validation part, the data is raw. The last explanation is about dropout. In training part, there will be dropout and the network will be loss some connections. However, in validation part, there is no dropout, therefore, the information is not loss, and the performance must be better than training part.

For the second problem, the main reason is that the convolutional neural network, Alexnet in this paper is not a very deep network. Therefore, some subtle characteristic cannot be extracted well. If the convolutional neural network can be changed as deeper network for example, GoogLeNet [9], the accuracy can be increased more.

4 Conclusion and Future Work

According to above experiments, the conclusion is that the accuracy of the convolutional neural network Alexnet is quite good when it is doing a 1362 classification task in the Vehicle-x dataset. This result shows that the model has had the value of application in real life if there are more improvement methods implemented.

There are also some future works to do. To prove the correctness of the above three explanations for the first problem can be done in the future. For example, researchers can change different methods to get the training loss and validation loss to verify whether the actual result is following the theory. When the researchers do not use any data augmentation methods and do not use dropout in training part, is the actual result is following the theory? If the answer is yes, it proves that the correctness of the second and third explanations above. The other future work is that if there are enough GPU resource in the future, larger batch size can be tested, and deeper convolutional neural network can be applied to increase the accuracy even this accuracy can up to state-of-the-art level.

References

1. Yao, Y., Zheng, L., Yang, X., Naphade, M., & Gedeon, T. (2019). Simulating content consistent vehicle datasets with attribute d escent. arXiv preprint arXiv:1912.08855.
2. Milne, L. K., Gedeon, T. D., & Skidmore, A. K. (1995). Classifying Dry Sclerophyll Forest from Augmented Satellite Data: Co mparing Neural Network, Decision Tree & Maximum Likelihood. training, 109(81), 0.
3. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.
4. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision (pp. 1026-1034).
5. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
6. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural n etworks from overfitting. The journal of machine learning research, 15(1), 1929-1958.
7. Smith, S. L., Kindermans, P. J., Ying, C., & Le, Q. V. (2017). Don't decay the learning rate, increase the batch size. arXiv preprint arXiv:1711.00489.
8. Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay. arXiv preprint arXiv:1803.09820.
9. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).