Cataract Classification using Inception, VGGNet, ResNet

Harmish Patel
School of Computer Science
University of Windsor
Windsor, Canada
patel2ch@uwindsor.ca

Khushkumar Harshadbhai Patel
School of Computer Science
University of Windsor
Windsor, Canada
patel2a4@uwindsor.ca

Sahil Sangani
School of Computer Science
University of Windsor
Windsor, Canada
sangan11@uwindsor.ca

***Abstract*—Cataract is the most widespread causes of blindness. Early detection or precautions could reduce the suffering from cataract to the patients and mitigate the visual disability from turning into total blindness. But the cost may cause difficulties to everybody's early interventions, because the expertise trained eye specialists cannot be afforded by everyone. Based on the data provided on Kaggle.com, we are trying to build a model that predict whether the patient is suffering from cataract or not.**

This paper aims to investigate the performance of three different model such as VGGNet, ResNet and Inception on the same dataset which contains the 4 classes for cataract detection, and comparison the result for the same models. While training models, will keep track of loss and accuracy of training vs validation and set the hyperparameters accordingly. As of now Inception has the best result with minimum loss and maximum accuracy among other two models.

Keywords—VGGNet, ResNet, Inception, cataract

I. Introduction

Cataract is a clouding of the lens in the eye that normally affects vision. Cataract, the most common cause of blindness and visual impairment, is often related to ageing. [1] More than 50% of cases of blindness is due to cataract. To overcome the issue of cataract and prevent visual impairment of blindness, early detection of cataract and relevant treatment is a must. Optometrists is the person who is responsible for diagnosis cataract detection, then further cataract surgery is done by the Ophthalmologists. However, there are only 300,000 optometrists globally, where half of them are located in developing countries, and we are in need of more than 1 million optometrists. [2] Moreover, the World Health Organization estimates that around 18 million people suffering from blindness due to cataracts. Unfortunately, every patient cannot afford the optometrists due the lack of resources and number of optometrists. There are mainly three types of cataract: Nuclear Sclerotic Cataracts, Cortical Cataracts, and Posterior Subcapsular Cataracts [3]. One of the major risk-factor for cataract is aging. Several other associated risk-factors are trauma, uveitis, diabetes, ultraviolet light exposure, and smoking. It is recommended early detected, early treated. To slow down the progress of cataract, surgical treatment is the best option. There are mainly four categories of cataract detection and grading: Light-focus method, Iris image projection, Slit lamp examination, and ophthalmoscopic transillumination. Manual assessment is still not as effective as it is subjective, time-consuming and costly. Therefore, there is need of automatic cataract detection using artificial intelligence emerges as it is acceptable from social and economic factors.

Image dataset can be categorised into four classes: Normal, Cataract, Glaucoma, and Retina Disease. Figure 1 illustrates sample images of each of four classes. Fig. 1(a) demonstrates the normal human eye with clear visible optic disc and blood vessels. Cataract images(b) shows only large vessels and blur optic disc or nothing can be visible. There are several structural changes in the optic nerve in Glaucoma image(c), whereas optic nerve fibre is almost damage in the case of retina disease(d).

Figure 1 (a): Normal Eye

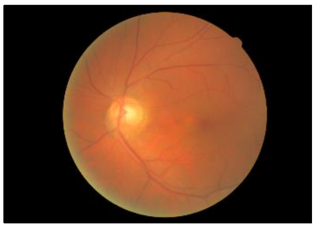

Figure 1 (b): Cataract

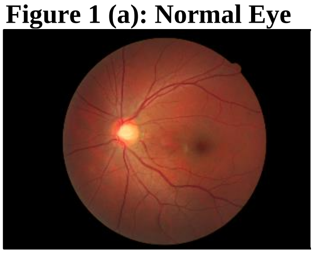

Figure 1 (c): Glaucoma

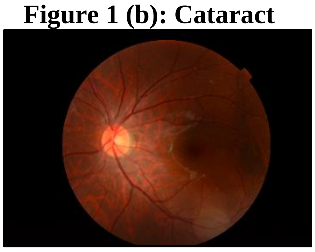

Figure 1 (d): Retina Disease

II. Related Work

Studies on image classification has been conducted for years. Basically, cataract classification mainly has four parts in it: pre-processing, feature extraction, feature selection and classifier. So, if there is any data, there must be noise; which leads to inconsistency. Pre-processing handles the noise as in few approaches such as: to enhance the condition of images, for instance, image improvement and noise removal. Segmentation and location of retinal structure, such as retinal lesions, vessels, optic discs, and aneurysms have been widely studied. To mitigate the difficulty of dimension mishap, feature selection selects the best possible dimension to reduce the complexity of dimensions. Feature representation consists of feature extraction and feature selection, which plays very critical role for the accuracy of the final model.

Understanding the representations learned by DCNN and observe learned features' invariance at different levels leads to high activations. Examinations of the effect of G-filter and the scalability of database on the DCNN classification accuracy, getting the accuracy of 93.52% on the dataset called retinal fundus images [4] from Beijing Tongren Eye Centre of Beijing Tongren Hospital. [5]

This paper focuses on comparing different pre-existing model such as VGG19 [6], ResNet [7] and InceptionV3 [8] on the same dataset as mentioned before and examine the results for the same.

III. Methodology

Classification of Cataract is made using various models of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) i.e. VGGNet, ResNet and Inception. Detailed architecture proposed by authors and layers description with various hyper parameters of each model is described in the further part of paper. In this comparison we used pre-trained models and using transfer learning we have updated final fully-connected layers to detecting the cataract. Architecture and layers details is as below for various models.

A. *Inception V3*

Inception v3 is a widely-used image recognition model that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset. The model is the combination of many ideas developed by multiple researchers over the years.

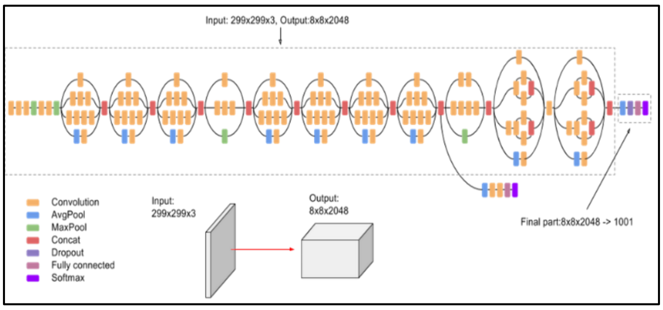

Figure 2: Inception V3 Model

The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, dropouts, and fully connected layers. Batch-norm is used extensively throughout the model and applied to activation inputs. Loss is computed via SoftMax.

Inception work with Factorizing Convolutions. Factorizing Convolutions used to reduce the number of connections and parameters to learn. This will increase the speed and gives a good performance.

GoogleNet [9] used a 5x5 convolution layer whereas in inception work with two 3x3 layers to reduce the number of learning parameters. In 5 x 5 has 25 total parameters were 3 x 3 + 3 x 3 has total 18 parameters to learn. So significantly no learning parameter is reduced by 28%.

Factorization into Asymmetric Convolutions is also used in Inception which also helps to help to reduce the learning parameter.

One 3×1 convolution followed by one 1×3 convolution replaces one 3×3. In one 3 x 3 has a total of 9 parameter whereas, 3 x 1 + 1 x 3 has a total of 6 parameter so it will reduce by 33%. This method is less likely to overfit the model as you go mode deeper in the training. [4]

With 42 layers deep, the computation cost is only about 2.5 higher than that of GoogleNet and much more efficient than that of VGGNet.

B. *ResNet*

The main base element of ResNet is the residual block. As we go deeper into the network with a large number of layers, computation becomes more complex. These layers put on top of each other and every layer try to learn some underlying mapping of the desired function and instead of having these blocks, we try and fit a residual mapping.

Here on this right where the input to these blocks is just the input coming in whereas on the other side, we're going to use our layers to try and fit some residual of our H(X) - X instead of the desired function H(X) directly. So basically, at the end of this block it takes the skip connection on this right here, where it just takes the input and pass it through as an identity, and so if it had no weight layers in between it was just going to be the identity. It would be the same thing as the output, but now we use additional weight layers to learn some delta, for some residual from our X.

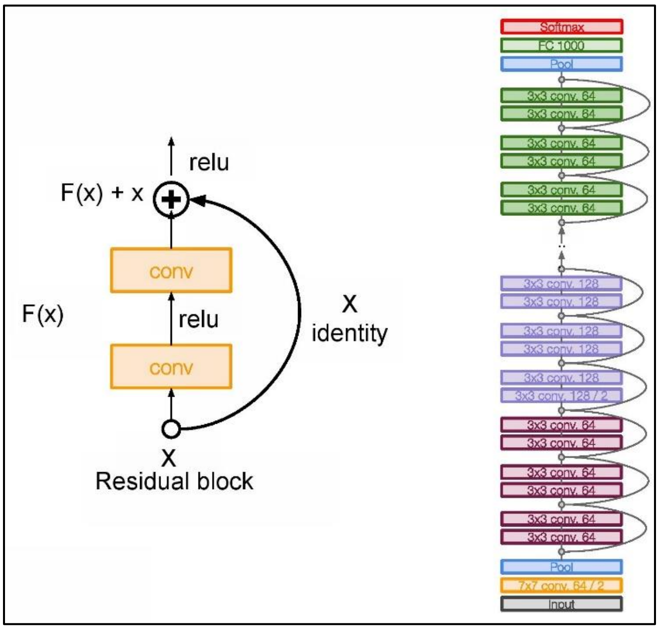

Figure 3: Residual Block

In nutshell, as we go deeper into the network it is so hard to learn H(X) as we have a large number of layers. So here we used skip connection and learning F(x) direct input of x as the final output. So, F(x) is called as a Residual.

In ResNet, stacks all these blocks together very deeply. Another thing with this very deep architecture is that it is enabling up to 150 layers deep of this, and then what we do is we stack all these layers periodically. We also double the number of filters and down-sample spatially using stride two. In the end, only fully connected layer 1000 to output classes.

In ResNet, it uses Batch Normalization after every conv layer. It also uses Xavier initialization with SGD + Momentum. The learning rate is 0.1 and is divided by 10 as validation error becomes constant. Moreover, batch-size is

256 and weight decay is 1e-5. The important part is there is no dropout is used in ResNet.

C. VGGNet

In 2014 there are a couple of architectures that were more significantly different and made another jump in performance, and the main difference with these networks with the deeper networks.

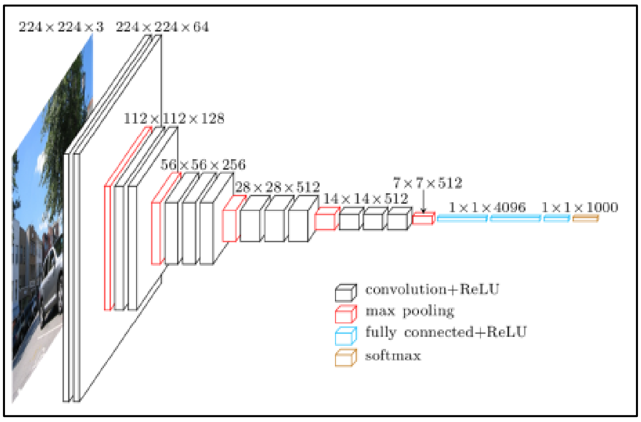

Figure 4: VGG Network

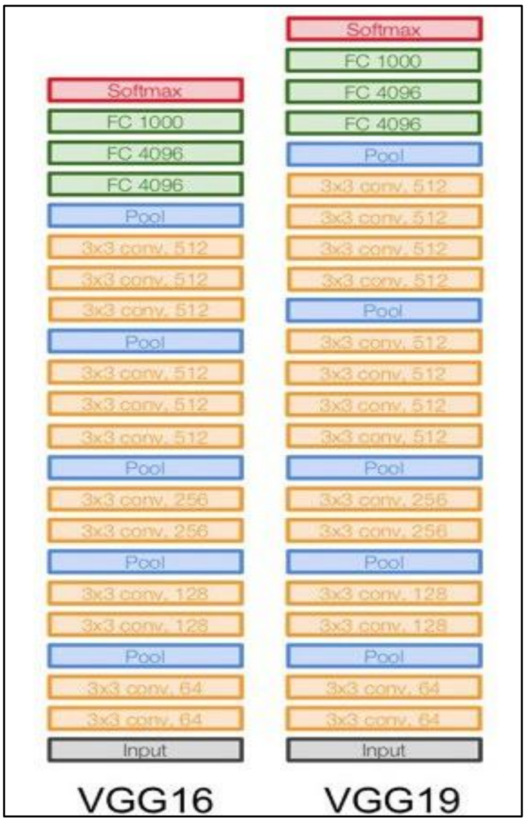

Figure 5: VGG16 vs VGG19

VGG 16 is 16-layer architecture with a pair of convolution layers, pooling layer and at the end fully connected layer. VGG network is the idea of much deeper networks and with much smaller filters. VGGNet increased the number of layers from eight layers in AlexNet. Right now, it had models with 16 to 19 layers variant of VGGNet. One key thing is that these models kept very small filters with 3 x 3 conv all the way, which is basically the smallest conv filter size that is looking at a little bit of the neighboring pixels. And they just kept this very simple structure of 3 x 3 convs with the periodic pooling all the way through the network.

VGG used small filters because of fewer parameters and stack more of them instead of having larger filters. VGG has smaller filters with more depth instead of having large filters. It has ended up having the same effective receptive field as if you only have one 7 x 7 convolutional layers.

VGGNet has conv layers and a pooling layer a couple more conv layers, pooling layer, several more conv layers and so on. VGG architecture has the 16-total number of convolutional and fully connected layers. It has 16 in this case for VGG 16, and then 19 for VGG 19, it's just a very similar architecture, but with a few more conv layers in there.

So, this is quite costly computations with 138M total Parameter and each image has a memory of 96MB which is so much large than a regular image. It has just a 7.3 error rate in the ILSVRC challenge.

IV. Implementation

A. Data Augmentation

The cataract dataset has various input sizes of the image. Images have different dimensions with different aspect ratios. It is important to make a uniform distribution of the image to pass it to the model for good accuracy. We have to use the PyTorch *[10]* data transform module to convert images into desired sizes. All the images have been passes with the random resized crop with its input size of the model. For VGGNet, ResNet and Inception have an input size of 224, 224, 299 respectively. So RandomResizedCrop has the same input size to crop every training image. Moreover, we have different images for the left and right eyes. So it is important to make it vertical flip of the image. So this will make good use of data with various varieties of fundus images of eye. Data Augmentation will help to create more data with a variety of inputs for training. Every training image is also normalized with a mean and standard deviation of the image. This step will make zero centered and normalized input in training.

Dataset contains images which has 4 classes in it; to reshape the image in a format in which image can be fed to the model so that model would perform best for particular dataset. For setting up image size, model requires the image size as a hyperparameter. There are other hyperparameters which affects the model during training as well as validating our model. Here are the hyperparameters for different model which describe in this paper:

B. Hyperparameters

1) Inception V3

Image input size: 299
No of classes: 4
Criterion: CrossEntropyLoss
Batch size: 16
Optimizer: SGD
Learning rate: 0.001
Momentum: 0.9
Epochs: 60

2) *ResNet*

Image input size: 224
No of classes: 4
Criterion: CrossEntropyLoss
Batch size: 16
Optimizer: SGD
Learning rate: 0.001
Momentum: 0.9
Epochs: 60

3) *VGGNet*

Image input size: 224
No of classes: 4
Criterion: CrossEntropyLoss
Batch size: 16
Optimizer: SGD
Learning rate: 0.001
Momentum: 0.9
Epochs: 60

Basically, we have used the same hyperparameters for almost all three models; Inception V3 requires the image size as 299 except that VGGNet and ResNet accept the size of image as 224.

"Cross-entropy loss measures the performance of classification models whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label." [11]

V. RESULT COMPARISON

To compare all three models, the loss for training and validation and the accuracy for the same training and validation is been monitored while feeding the data to the models, the result comparisons are as below:

A. *Inception V3*

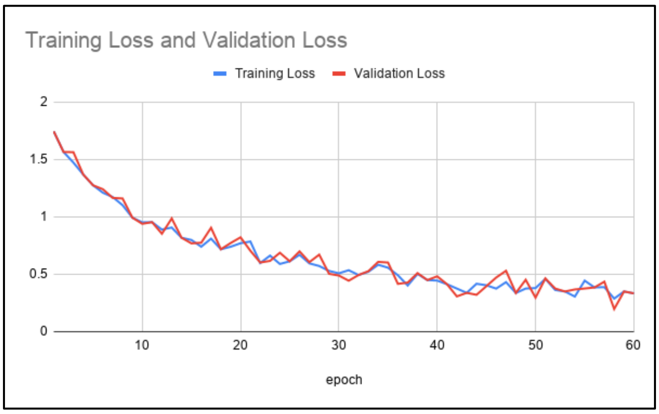

Figure 6: Training and Validation Loss vs epoch (Inception)

The loss in training and validation set is much higher in first 10 epochs for the inception model. Then, it constantly dropped as number of epochs increased. It finally ended with 0.32.

In the Inception model, an accuracy of training set and validation set is almost the same as shown in the Figure 7**.** Training accuracy is 90%, whereas validation accuracy achieved almost 87%.

From figure 7 we can observe that model is learning in starting epoch of training. But after reaching epoch 43 it is becoming almost constant. As training accuracy is improving but validation accuracy remains constant with the training of the epochs.

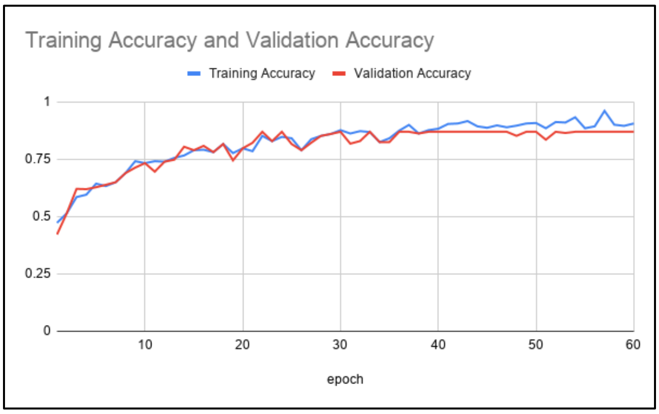

Figure 7: Training and Validation Accuracy vs epoch (Inception)

After epoch 43 model learns so slowly till 56 epochs. Training more will lead to overfitting problem as training accuracy is improving but validation accuracy remains constant with the time. So, to reduce overfitting we have stopped after 60 epochs.

B. *ResNet*

Training and validation loss for ResNet model shown in the below Figure 8. It is observed that it is more optimized model compare to VGG as loss overcome 0.14.

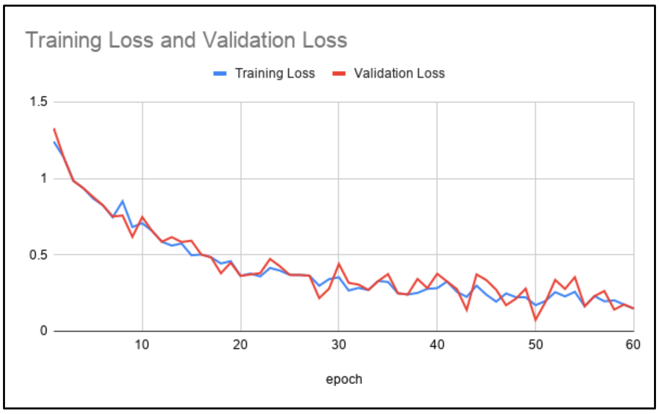

Figure 8: Training and Validation Loss vs epoch (ResNet)

Below Figure 9 represents graph of training and validation accuracy versus number of epochs for ResNet model. It can be observed that graph is smoother and gap between training accuracy and validation accuracy remains consistent over time.

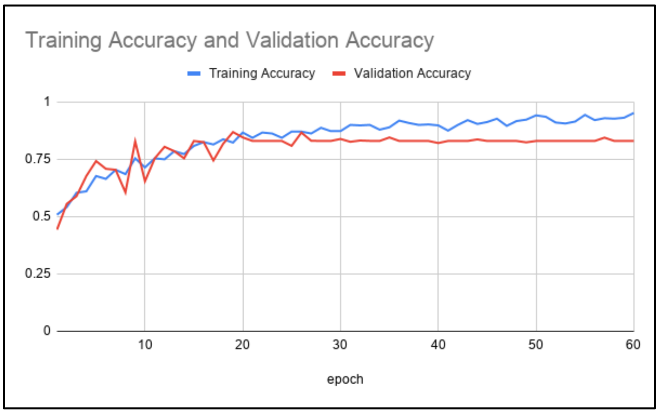

Figure 9: Training and Validation Accuracy vs epoch (ResNet)

In compare to Inception ResNet start overfitting early with epoch 40 but still its overall accuracy is less than Inception and model is overfitting with more training.

C. VGGNet

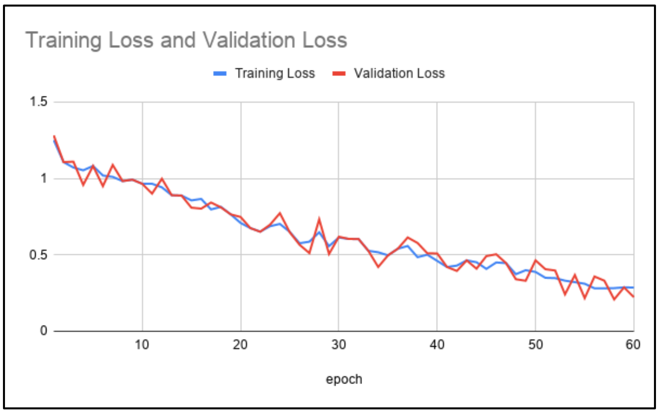

Figure 10: Training and Validation Loss vs epoch (VGGNet)

As shown in Figure 10, It demonstrates the training and validation loss over the number of epochs. It can be seen that both the losses gradually decrease as the number of epochs are increases, and ended up with 0.32.

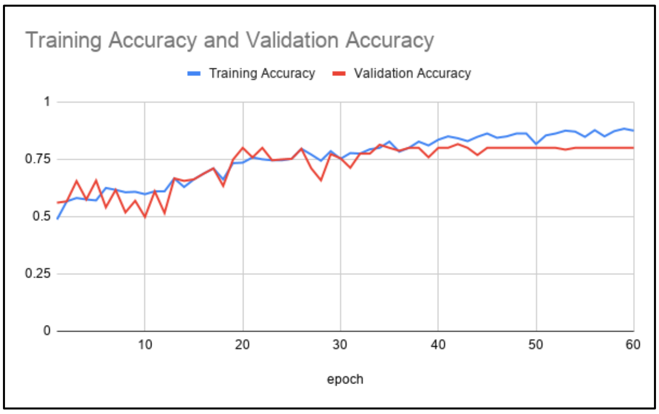

Figure 11: Training and Validation Accuracy vs epoch (VGGNet)

As illustrate in Figure 11, The training and validation accuracy versus number of epochs. As shown in the figure, it can be visible both started with 50% to 60% accuracy. Further, it increases as increase of number of epochs and ended with almost 87% accuracy.

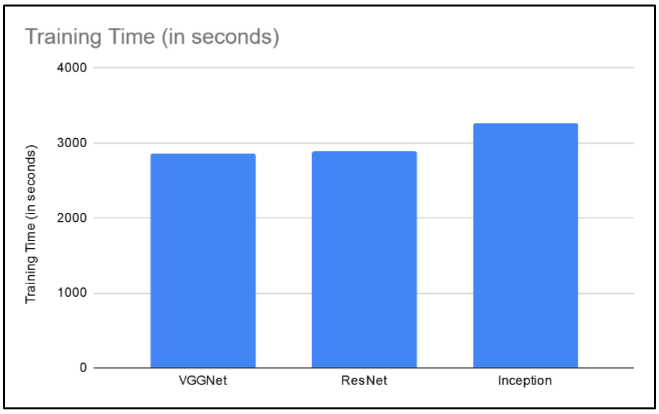

Figure 12: Training Time for different models

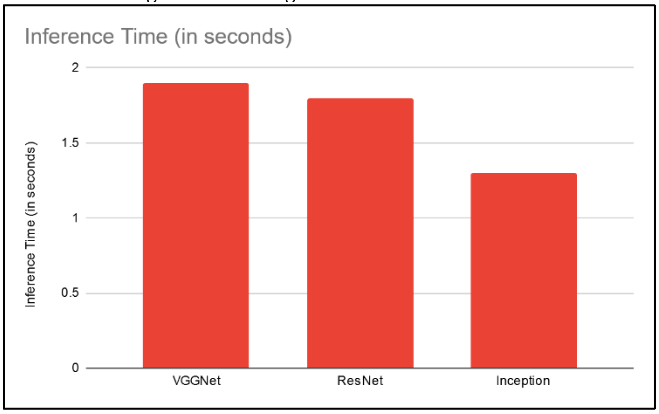

Figure 13: Inference Time of different models

Training time of VGGNet, ResNet, and Inception is 2867, 2984 and 3270 seconds respectively. It clearly shows that training time for Inception is 286 seconds more but inference time of Inception is exceptionally low comparatively ResNet and VGGNet. Inference time is more important because model training is once where inference is frequently so Inception also better with the performance.

VI. Research Questions

While comparing three models on dataset we encountered certain things which should be research further and could be mitigate while implementing other models. Following are the questions:

A. *What are the changes made in hyperparameters, which leads to lower the loss and to higher the accuracy?*

As paper focuses on the comparison of three models and all three have some default hyperparameters and boundaries which needs to satisfied. i.e. Inceptions V3 needs image size of 299 and other two models require image size of 224. Dataset has images which has different image size and models requires the uniform sizes; so, changes in image size is essential. Other than image size, batch size, epoch, learning rate are same among all three models. So that comparison could be made.

B. *What are the suggestion to imporve the accuracy to classify the cataract?*

Currently, to classify cataract images we are using pre-trained models of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) i.e. VGGNet, ResNet and Inception. These models are trained on 1000 real-world objects. To classify the cataract, we are fine-tuning it and used transfer learning. Improvement in this accuracy is done if we implement transfer learning in a pre-trained model of retina or eye images. This will give good insights and improvement in accuracy as well as misclassification rate will decrease. Another solution is also that we can design our own neural network to classify cataracts. It will require a high amount of data which is also a bottleneck for this project. Currently, we just have 601 combined images of 4 classes which is quite low to train the neural network. We can train model from scratch as we have a greater number of images.

Conclusion

In nutshell, among ImageNet Large Scale Visual Recognition Challenge (ILSVRC) models i.e VGGNet, ResNet and Inception, inception is performing better with same hyper parameters. Inception V3 has total 152 layers which is much deeper network comparatively ResNet and VGGNet. This will take some more time for training but inference time for classification of cataract classes is very efficient and less. In addition, accuracy is quite higher with 87% on validation set which shows good classification of cataract disease. Inception is performing better because it uses factorization and auxiliary classifier which gives more insights about classification of different classes. In terms of inference time, it is performing excellent because it has 28% less learning parameters than ResNet without decrease the network efficiency.

References

[1] WHO, "What is cataract?," 2019. [Online]. Available: www.emro.who.int/health-topics/cataract/introduction.html#:~:targetText=Cataract is a clouding of,is often related to ageing.&targetText=Sometimes%2C the development of cataract,to some types of radiation.. [Accessed 27 November 2019].

[2] Healio, "Global need for vision care indicates need for more optometrists," August 2012. [Online]. Available: https://www.healio.com/optometry/primary-care-optometry/news/print/primary-care-optometry-news/%7B5dee8831-92d1-49bf-9108-f8457cd4cf6e%7D/global-need-for-vision-care-indicates-need-for-more-optometrists#:~:targetText=Luigi%20Bilotto%2C%20OD%2C%20of%20the,. [Accessed 27 November 2019].

[3] American Printing House, "Are There Different Types of Cataracts?," 2019. [Online]. Available: https://www.visionaware.org/info/your-eye-condition/cataracts/different-types-of-cataracts/125#:~:targetText=There%20are%20three%20primary%20types,types%2C%20can%20develop%20over%20time.. [Accessed 27 November 2019].

[4] Kaggle Inc, "cataract dataset," September 2019. [Online]. Available: https://www.kaggle.com/jr2ngb/cataractdataset. [Accessed 30 November 2019].

[5] L. a. L. J. a. H. H. a. L. B. a. Y. J. a. W. Q. a. o. Zhang, "Automatic cataract detection and grading using Deep Convolutional Neural Network," *IEEE 14th International Conference on Networking, Sensing and Control (ICNSC),* pp. 60--65, 2017.

[6] K. a. Z. A. Simonyan, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[7] K. a. Z. X. a. R. S. a. S. J. He, "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 770--778, 2016.

[8] C. a. V. V. a. I. S. a. S. J. a. W. Z. Szegedy, "Rethinking the inception architecture for computer vision," *Proceedings of the IEEE conference on computer vision and pattern recognitio,* pp. 2818--2826, 2016.

[9] C. a. L. W. a. J. Y. a. S. P. a. R. S. a. A. D. a. E. D. a. V. V. a. R. A. Szegedy, "Going deeper with convolutions," *Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 1--9, 2015.

[10] Torch Contributors, "torchvision.transforms," 2019. [Online]. Available: https://pytorch.org/docs/stable/torchvision/transforms.html. [Accessed 9 December 2019].

[11] Revision 730d1a0c, "Loss Functions," ML Glossary, 2017. [Online]. Available: https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html. [Accessed 5 December 2019].