Assignment2: Convolution (Cat & Dog)

Name: Kandarp Barot

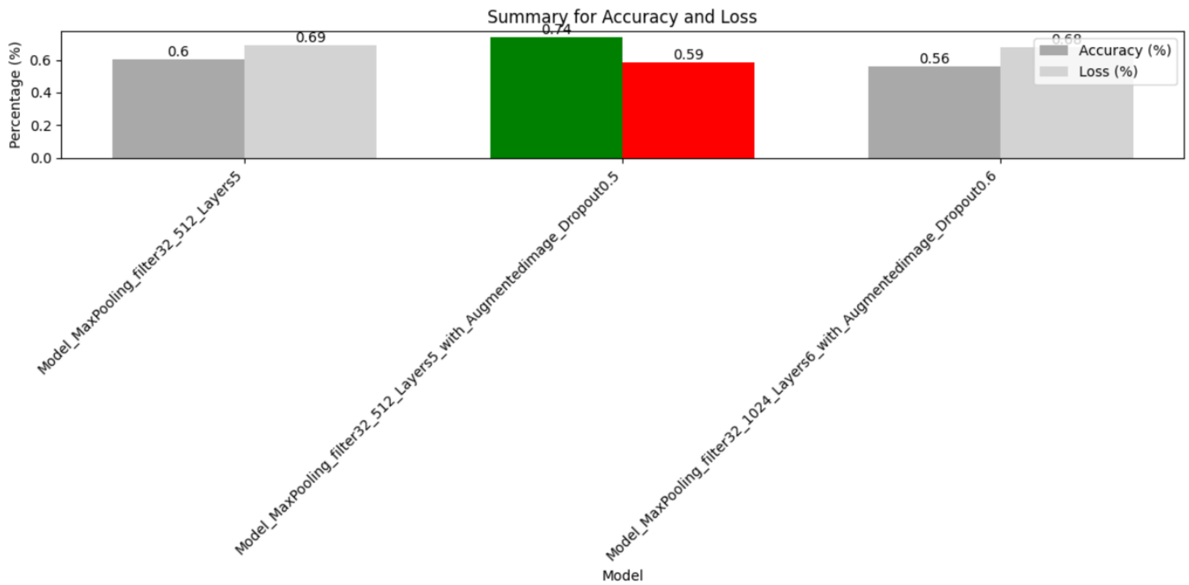
Date: 7/14/2024

In this assignment of we did try to build total 11 models with the training sample being 1000, 2000, 3000 and 5000. Also tried to reduce overfitting by using various methods. Compared the model build from scratch vs pre-trained VGG16 Model.

Below we will Discuss the loss and Accuracy of all the models to see which model gives better result and why. And will provide the Recommendations.

Question 1: Consider the Cats & Dogs example. Start initially with a training sample of 1000, a validation sample of 500, and a test sample of 500 (like in the text). Use any technique to reduce overfitting and improve performance in developing a network that you train from scratch. What performance did you achieve?

Answer: Below is the graph representing the loss and accuracy of all the 3 Models created by changing the parameters and sample size 1000.

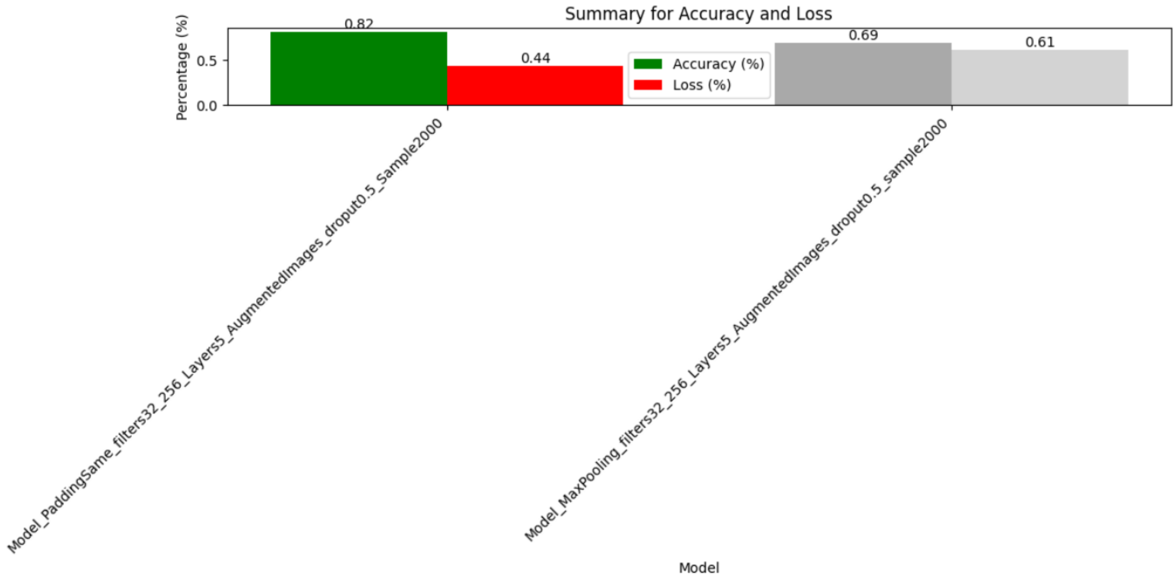


From the above graph we can conclude that model with Max-pooling with layer 5, increasing filters from 32 to 512 with image augmentation is the best among all with higher accuracy and minimum loss, however 3rd model has lowest accuracy and highest loss because of increasing the filters to 1024 with 6 layers.

As we can see 2nd model is performing best, we should choose model with filters from 32 to 512, 5 Input Layers, Augmented Images and Dropout rate of 0.5

Question 2: **Increase your training sample size. You may pick any amount. Keep the validation and test samples the same as above. Optimize your network (again training from scratch). What performance did you achieve?**

Answer 2: To address the second query, we did try to build 2 more models with training samples being 2000. We plotted the 4th and 5th Model, allowing a visual comparison of their performance. The graphs distinctly illustrate that the 4th model “max pooling with padding” achieved the highest accuracy among all the models, reaching 84%, with the lowest loss of 45.9%. The augmentation of the training samples to 2000 and the introduction of diverse augmented images notably contributed to the improved performance of them. Now let’s compare the loss and Accuracy of these models with previous model to see which model gives better result.



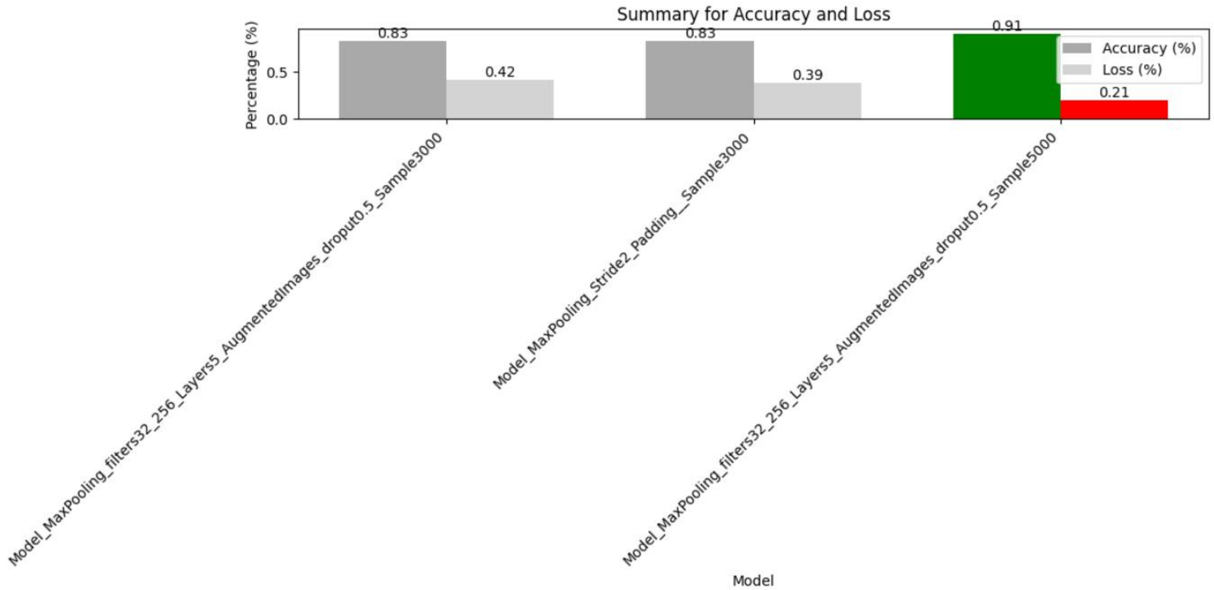
The second model built just with 1000 training samples resulted in 74% accuracy whereas the same model with a further increase in training samples to 2000 spiked the accuracy to 82% i.e. 9% increase in the accuracy.

Comparing the performance of the models, it was observed that utilizing strides with padding did significantly benefit the model. The model, incorporating a Max Pooling with padding Layer, exhibited a 09% higher accuracy compared to the model without padding. Furthermore, by optimizing the network and augmenting the training dataset from 1000 to 2000 samples, an enhanced accuracy of 82% was achieved.

Question 3: Now change your training sample so that you achieve better performance than those from Steps 1 and 2. This sample size may be larger, or smaller than those in the previous steps. The objective is to find the ideal training sample size to get best prediction results

Answe3: In the previous 4^{th} Model, we attempted to replace the conventional max pooling operation with strides, the results were promising as expected. and in the 5th model we used max pooling only. Therefore, we are exploring a hybrid approach that combines both max pooling and strides to evaluate the performance of this new model.

Let's see which of the models have best performance when the training sample was set to 3000. Note: Here 7th model was trained differently with strides being used with max-pooling



Max pooling is a down sampling operation that reduces the spatial dimensions of the feature map, aiming to capture the most prominent features while discarding less relevant information. On the other hand, strides determine the step rate of the sliding window used to extract and learn the features from the data. This hybrid approach aims to leverage the advantages of both techniques, potentially enhancing the model's ability to capture intricate patterns and features while maintaining computational efficiency.

We constructed 3 models, 2 of which were trained with a sample size of 3000. The top-performing model achieved 83% accuracy. Notably, when we expanded the training sample to 5000, the accuracy rose to 91%. Consequently, we deduce that increasing the training sample to 5000 substantially enhances the model's performance. Regarding the plausible reason for the validation loss being lower than the training loss, it is likely influenced by the adopted split strategy. In this case, the training sample is nearly as extensive as 5000, while the validation and test sets remain fixed at 500 each. Additionally, it is essential to acknowledge that during training, regularizations such as dropout or L1 and L2 regularizes play a significant role, contributing to the computation of the training loss. Conversely, during the validation or test phase, these regularizes are disabled, potentially leading to a lower loss compared to the training loss.

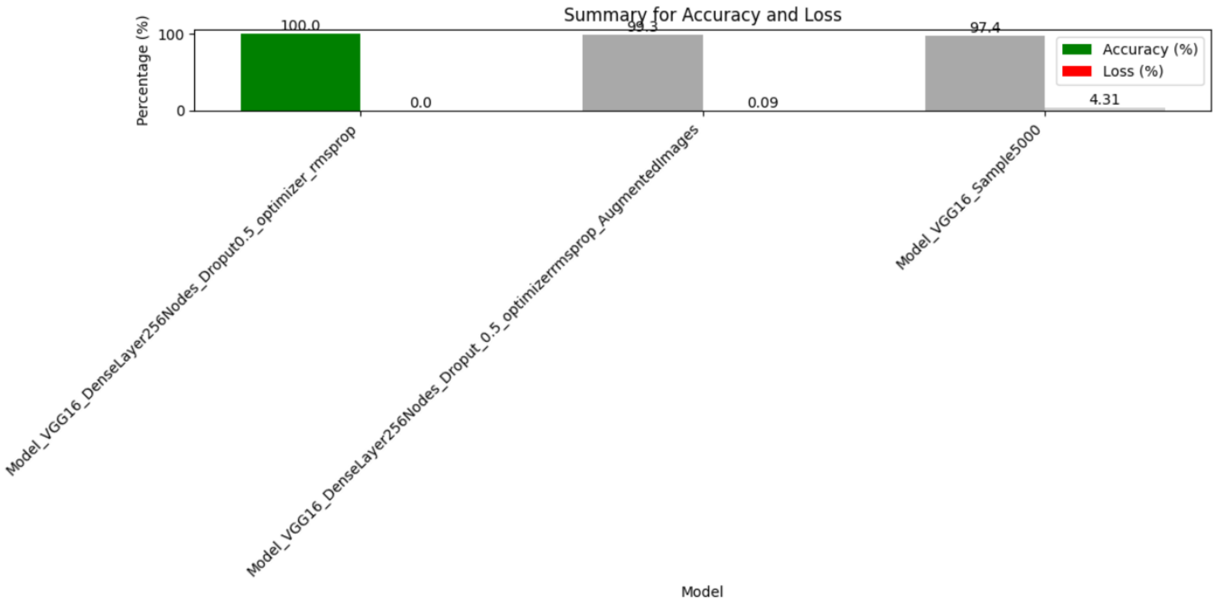
Question 4: Repeat Steps 1-3, but now using a pretrained network. The sample sizes you use in Steps 2 and 3 for the pretrained network may be the same or different from those using the network where you trained from scratch. Again, use all optimization techniques to get the best performance.

Answer 4: We have configured the pre-trained model to retain its existing weights during training, only allowing the densely connected networks and the classifier to adjust their weights during the training process.

This approach helps prevent overfitting as the pre-trained model remains unchanged, providing a stable foundation for the model. Moreover, when dealing with limited training data and constrained computational resources, freezing the pre-trained model training can be particularly advantageous.

It is crucial to understand that pre-trained networks are not solely utilized for singular image classification tasks; rather, they are trained to handle diverse use cases and classifications. The initial layers of the network are adept at capturing general features, while the subsequent layers tend to specialize in extracting features specific to the problem at hand. When we opt to freeze the initial layers, we effectively prevent overfitting, enabling the model to assimilate more intricate details pertaining to our specific classification task. This strategic approach encourages the model to focus on learning the nuanced aspects of the target classification problem

In the VGG16 models analysis, we constructed three models utilizing the pre-trained network VGG16. Notably, we observed that preventing the pre-trained network from updating its weights during training, and subsequently freezing the initial layers of the pre-trained network, resulted in higher accuracy. Consequently, we aim to employ the same methodology in building models using a training sample size of 5000.



Now, having constructed a total of 11 models, we are poised to conduct a comparative analysis to determine the best-performing models in two distinct categories: Scratch Models and Pre-Trained Models. Our immediate focus is to evaluate the best model developed from scratch. This evaluation entails comparing the loss and accuracy metrics of the 11 models built across four different training samples. The primary objective is to ascertain the optimal training sample size for the task of classifying cats and dogs

Conclusion:

The accuracy of a model is closely linked to both the volume of training data and the underlying architecture, especially when the model is trained from scratch. In contrast, when using a pretrained model, accuracy depends on the specific set of test samples being evaluated. Some sample sets may be more challenging than others, and strong performance on one set may not generalize to all sets.

Scratch Model: The size of the training sample and the model architecture significantly impact test accuracy. Incorporating modern architectures like residual connections, batch normalization, and depth wise separable convolutions into a basic scratch model, along with data augmentation and dropout techniques, substantially increases test accuracy. Increasing the training data size from 1,000 to 3,000 samples leads to notable improvements in test accuracy. Further expanding the dataset to 5,000 samples results in test accuracy levels comparable to those of pretrained models. This improvement is due to overfitting, where a small number of samples limits the model's ability to generalize. A larger dataset exposes the model to a broader data distribution, enhancing test accuracy.

Pretrained Network: When using a pretrained network, the highest test accuracy is achieved, but it is not significantly affected by the size of the training sample. This is because the model used to measure accuracy is not trained with the data it processes. Additionally, due to the large size of the dataset used in the pretrained VGG16 model (over 138 million parameters and more than 500 MB in size), techniques like data augmentation and fine-tuning do not significantly enhance accuracy, as the original pretrained model already has high-performance accuracy.

Recommendations:

The accuracy of the simple scratch model saw a significant increase, rising from 56% to 74%, when the training data was expanded from 1,000 to 3,000 samples. With 3,000 training samples, incorporating three modern architectures, along with data augmentation and dropout techniques, resulted in a test accuracy of approximately 82.2%. Furthermore, training the model with 5,000 samples achieved a test accuracy of 91%. Therefore, it is recommended to prioritize the scratch model with a sample size of 5,000 due to its superior accuracy.

For the pretrained fine-tuned model, we achieved 100% accuracy. However, I believe that pretrained models are not always the best choice for every situation. In the task of distinguishing between cat and dog pictures, the context or background of the image does not affect the distinction. Therefore, a pretrained model that distinguishes images can be effectively used in any image classification task, provided the sample images belong to the same category as those used to train the model.