**Summary**

The present case involves the division of a part of our dataset into training, testing, and validation datasets. Our model includes several layers; for this particular use, we employ Convolution2D layer, which is more appropriate for handling 3D images or videos.

Max pooling layer is used to reduce the spatial dimensions Finally, and we add flatten layer to convert the data into 1-D vector and the last layer is the Dense layer with sigmoid activation function.

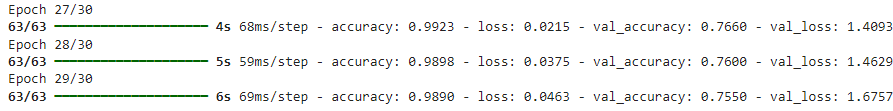
Reason for max pooling:

Failure of pooling layer is to reduce the size of feature maps to minimize the number of parameters which requires to be learned and number of operations which needs to be performed in the network. In doing so pooling facilitates the search for features invariant with respect to certain variations and acts as a constraint that reduces the likelihood of overfitting.

So, I performed basic convolutional operations on the trained dataset, but noticed problems of overfitting. The current issue remains, however, within the Deep Learning framework and more zone-in on Convolutional Neural Networks; how do we avoid overfitting?

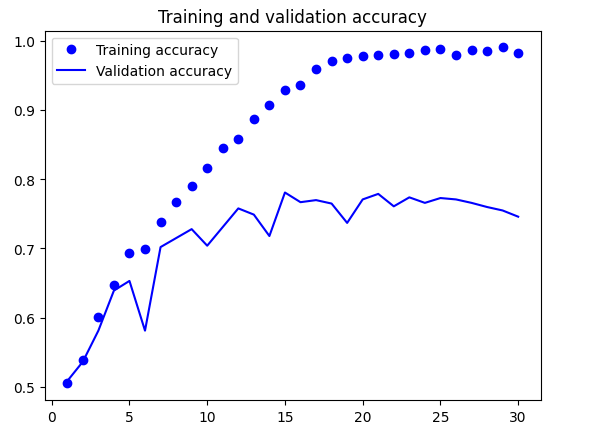
* First incorporate additional data into training data set
* Implemented regularization and optimization algorithms and used dropout techniques as done previously in assignment 1.
* Cross-validation technique is implemented.
* Using data augmentation method for better performance
* Adding noise to input data

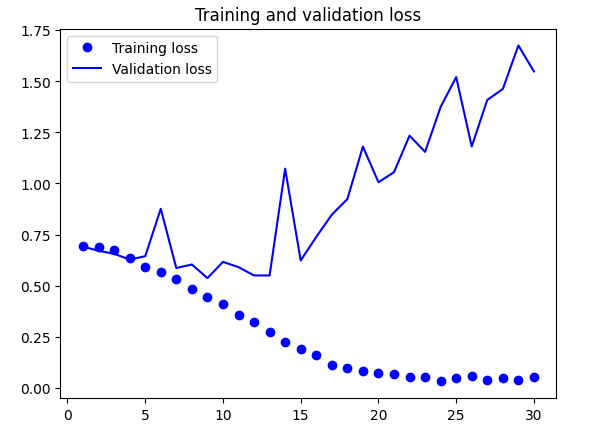
Initial Approach with basic convent and step 1



Observed Accuracy of 98% and loss of 0.046, validation accuracy of 75% and validation loss of 1.67

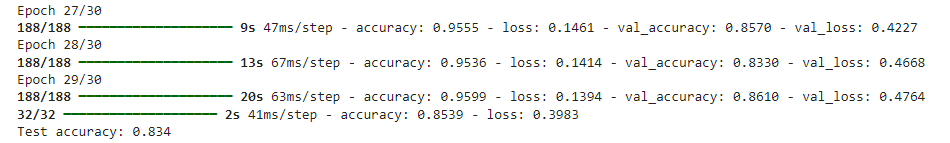
Now the below plotting shows the Training and validation accuracy and loss



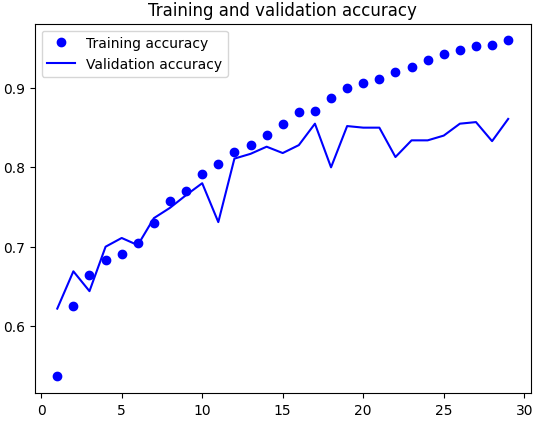


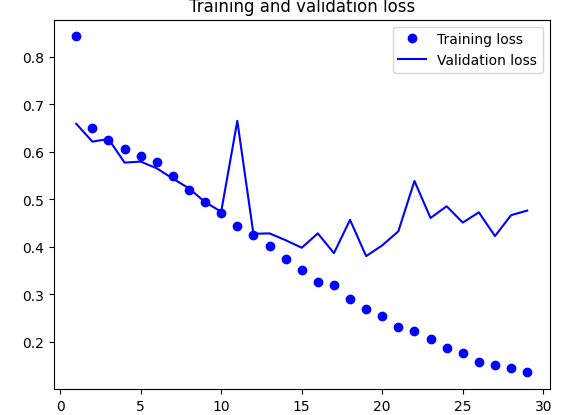
Increased training samples and performed again as this method enhanced the accuracy and validation some better performance was observed this led to reduction in overfitting.

**Step 2:**



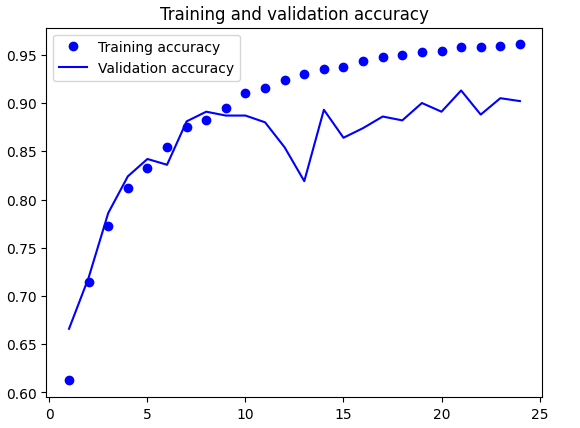
As we observed from the above by increasing the sample size of training set accuracy and validation accuracy started at 50% and 60% and increased to 95% and 86%. Used various techniques for augmenting data, rotation, and random flipping along with dropout, max pooling and convolution. But we can’t observe the performance as there is no betterment in model.

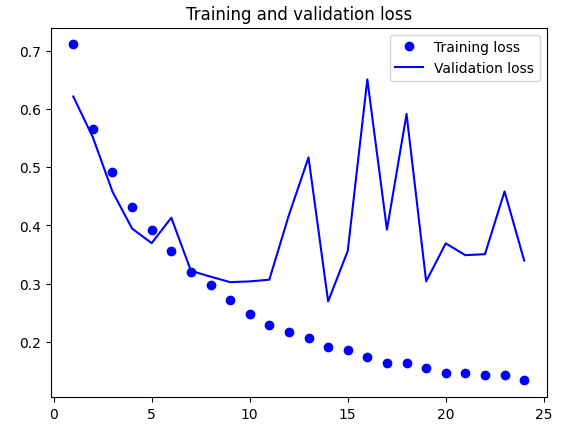




Step 3:

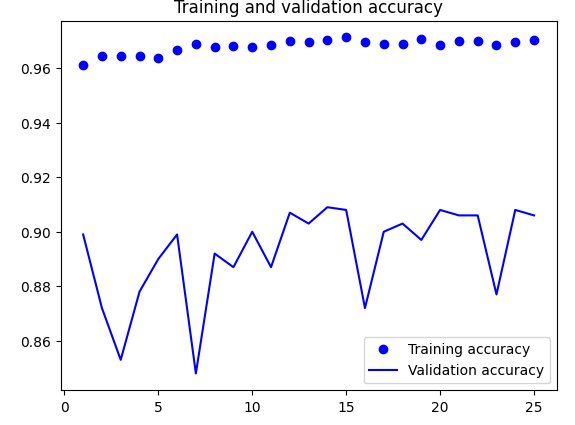
In this step convolutional neural network and architecture and augmenting data was used for better performance. Observed a test accuracy of 90% training accuracy 96%. We can observe the tuned model performance in this step and can see the prevention of overfitting. This model performed better in the image classification task, by acquiring the better accuracy for training and testing datasets.

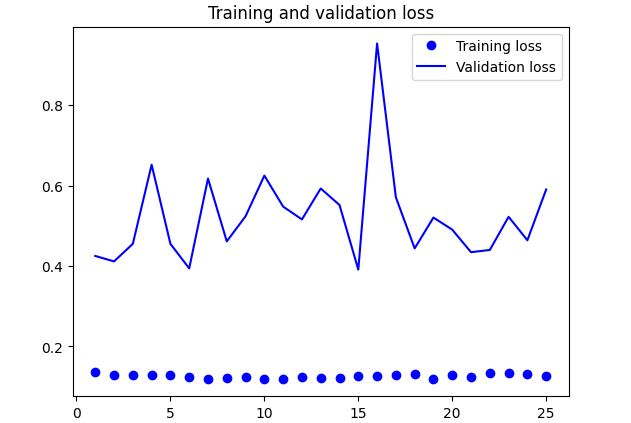




Step 4:

Loaded VGG16 Model. Pretrained on ImageNet dataset and tuned for new dataset. Used early stopper in this model to optimize overfitting. The training accuracy and validation accuracy reached to 97% and 90%. This model has a balanced performance and has high generalization.

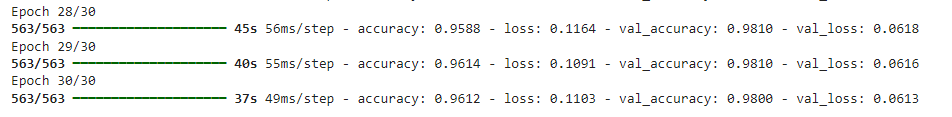




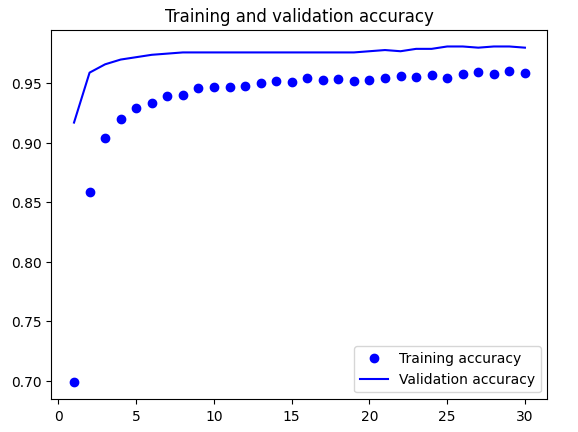
Pretrained Model 2:

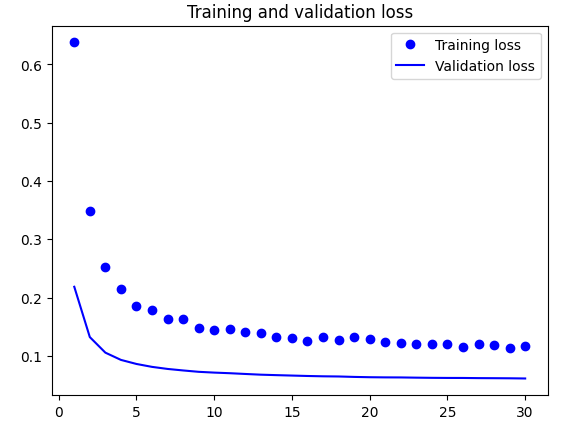
Used ResNet50V2 convolutional neural network with Tensorflos keras to classify photos of dogs and cats. Used ReLu activation and sigmoid function for binary classification and Adam optimizer for training the model. Observed a validation accuracy of 68% and accuracy of 97% there is a overfitting in this model which hasn’t happened in previous models.

Pretrained model 3:



This is the final layer of optimization based on MobileNetv2 convolutional base. Here I have considered 30 epochs in order to avoid overfitting. Here in this model we can observe a accuracy of 96% and validation accuracy of 98% the model performed well in this approach. Accuracy obtained was high with the use of data augmentation and transfer learning.





Overall Accuracy Observed throughout the CNN model:

Initial Model: Training Accuracy -97%, Val Accuracy-74% and Test Accuracy 70%

CNN Model with increased samples: Training Accuracy- 95%, Val Accuracy – 86% and Test Accuracy 83%

Optimized CNN Model: Training Accuracy – 96% , Val Accuracy 90% and Test Accuracy 90%

Pretrained Model 1: Training Accuracy – 97% and Val Accuracy 90% and Test Accuracy of 91%

Pretrained Model 2: Training Accuracy – 97% and Validation Accuracy 68% and Test Accuracy of 67%

Pretrained Model 3: Training Accuracy – 96% Validation Accuracy 98% and Test Accuracy of 98%

Overall in this assignment learned the impact of the training sample size and the choice of the network architecture on the two-class image classification task. It was found out that when training CNN models from scratch, high training accuracy was obtained although there was striking overfitting. Validation accuracy was improved by optimizing the network and size of the sample used in the training section. Applying the pretrained models (ResNet50V2, MobileNetV2, and VGG16) produce better accuracies for the proposed models showing the importance of transfer learning in image classification.