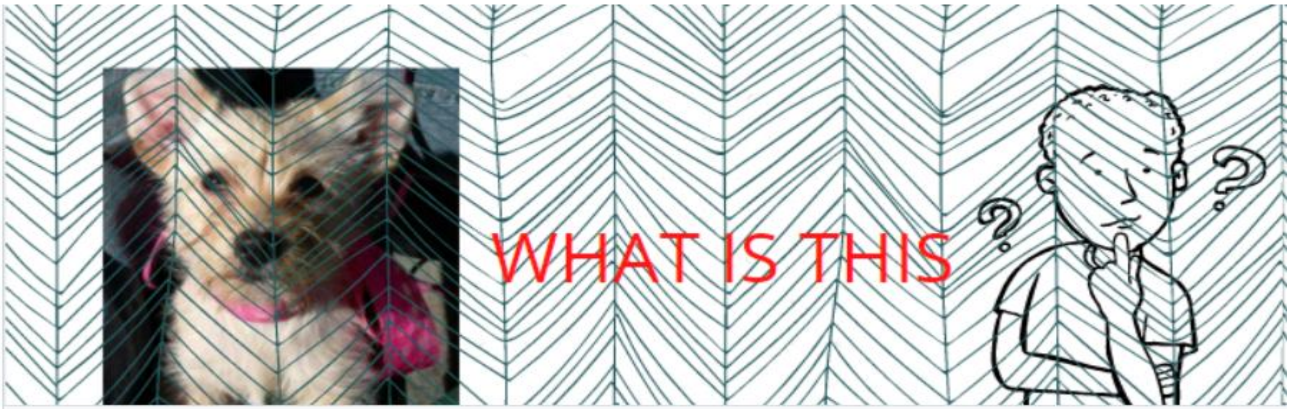
CONVOLUTIONAL NEURAL NETWORK

Image Classification



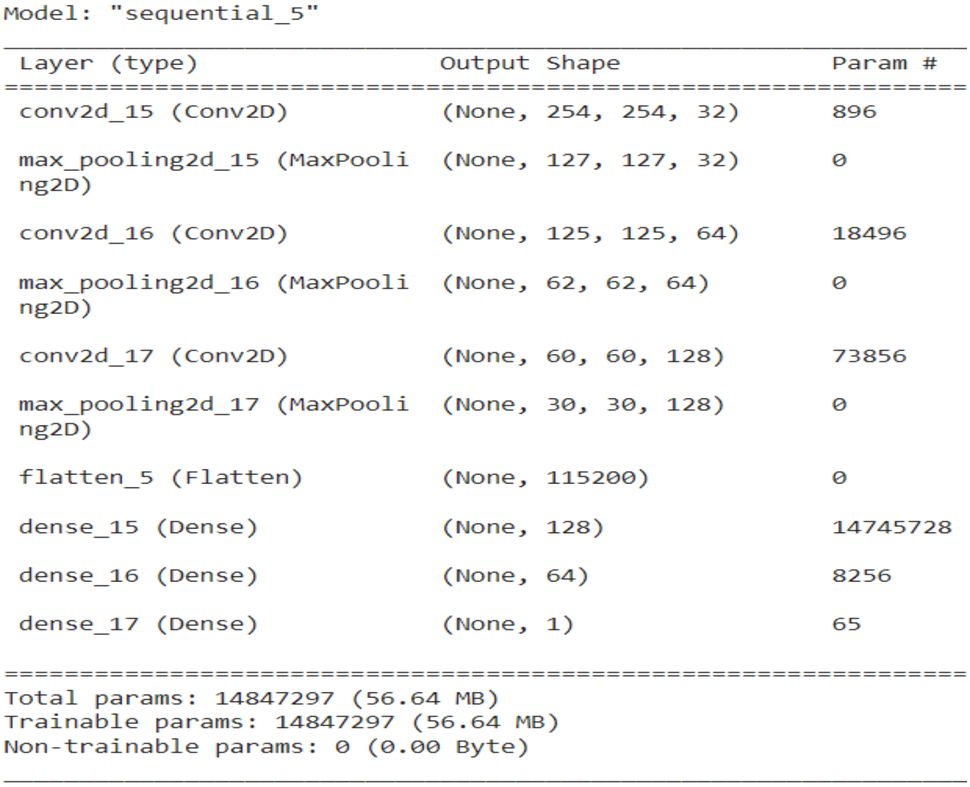
Objective :- CATS vs DOGS Classification using Convolutional Neural Networks and Data Augmentation

Data Source :- https://www.kaggle.com/datasets/salader/dogs-vs-cats

The dataset I'll be using consists of numerous images classified into two categories: cats and dogs. Each image has been labeled to facilitate a binary classification task. These images provide a rich dataset for training the classifier.

Model 1 Summary:

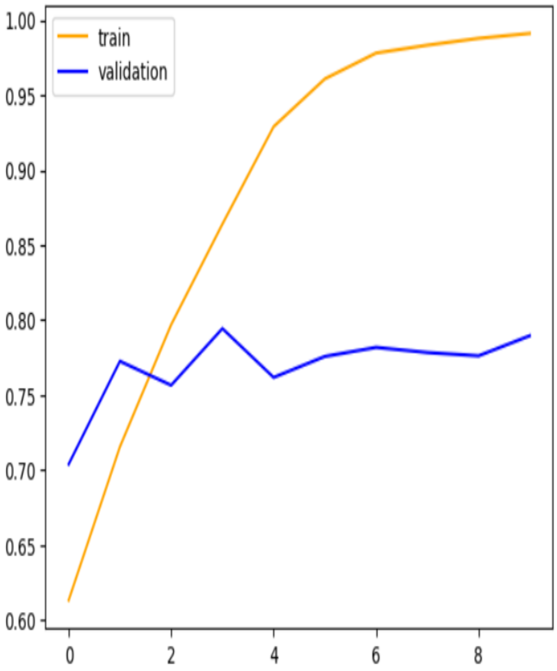
I used convolutional neural networks with 32, 64 and 128 layers.



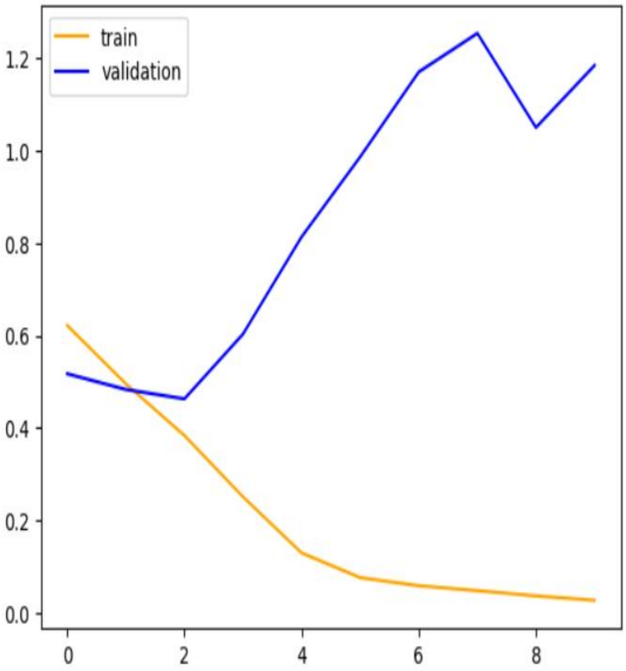
```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_15 (Conv2D)          (None, 254, 254, 32)      896

 max_pooling2d_15 (MaxPooli  (None, 127, 127, 32)      0
 ng2D)

 conv2d_16 (Conv2D)          (None, 125, 125, 64)      18496

 max_pooling2d_16 (MaxPooli  (None, 62, 62, 64)        0
 ng2D)

 conv2d_17 (Conv2D)          (None, 60, 60, 128)       73856

 max_pooling2d_17 (MaxPooli  (None, 30, 30, 128)       0
 ng2D)

 flatten_5 (Flatten)         (None, 115200)            0

 dense_15 (Dense)            (None, 128)               14745728

 dense_16 (Dense)            (None, 64)                8256

 dense_17 (Dense)            (None, 1)                 65

=================================================================
Total params: 14847297 (56.64 MB)
Trainable params: 14847297 (56.64 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Training and Validation Graph:

Accuracy



Loss function



Results:
Achieved 80% Accuracy on Validation data.

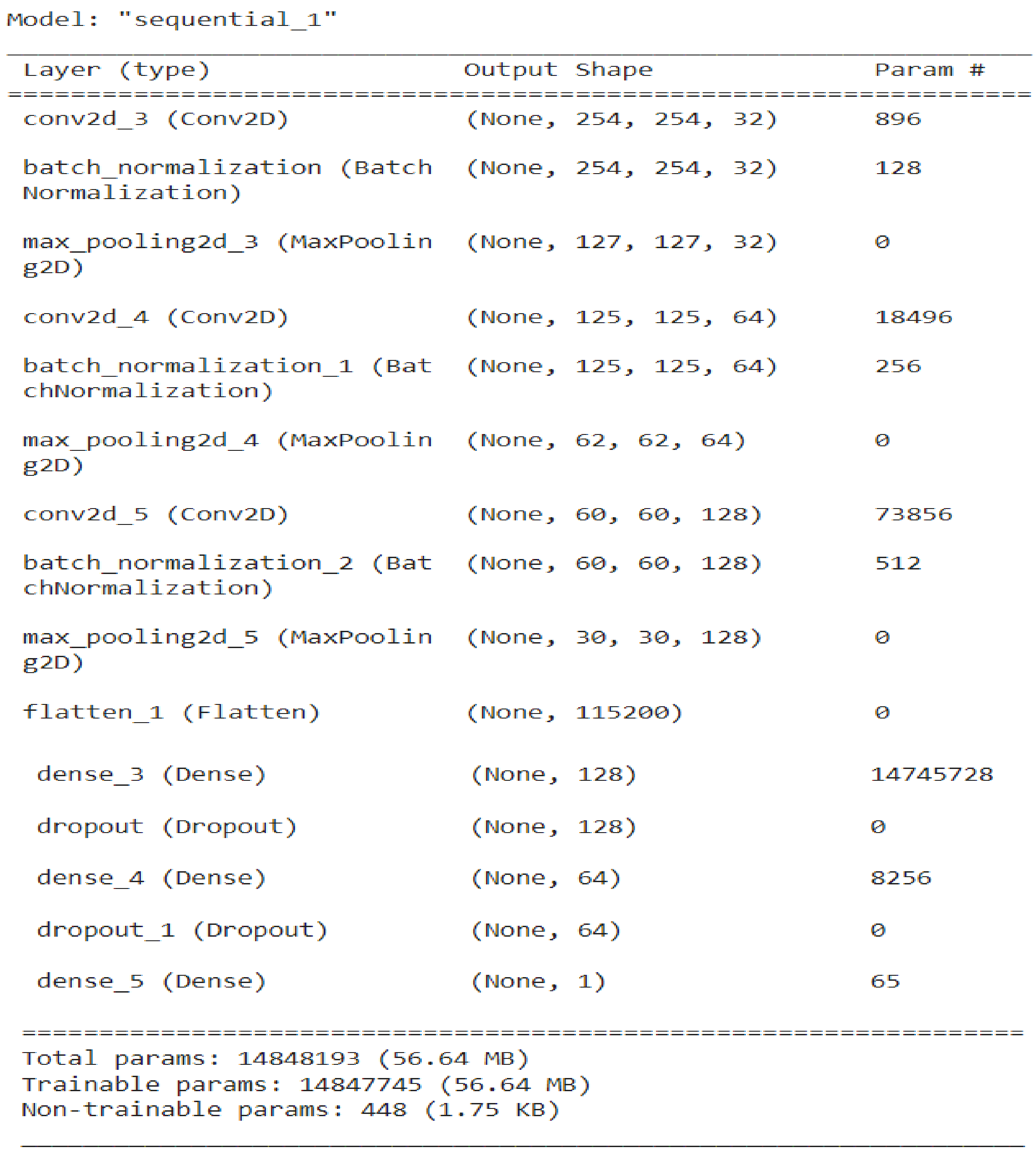
As we can see that our model is performing well on train data but not performing well on Validation Data that indicates that there is overfitting occurring.

Model 2

These are the ways to deal with overfitting problem :

1. Data Augmentation
2. L1/L2 Regularizer
3. Dropout

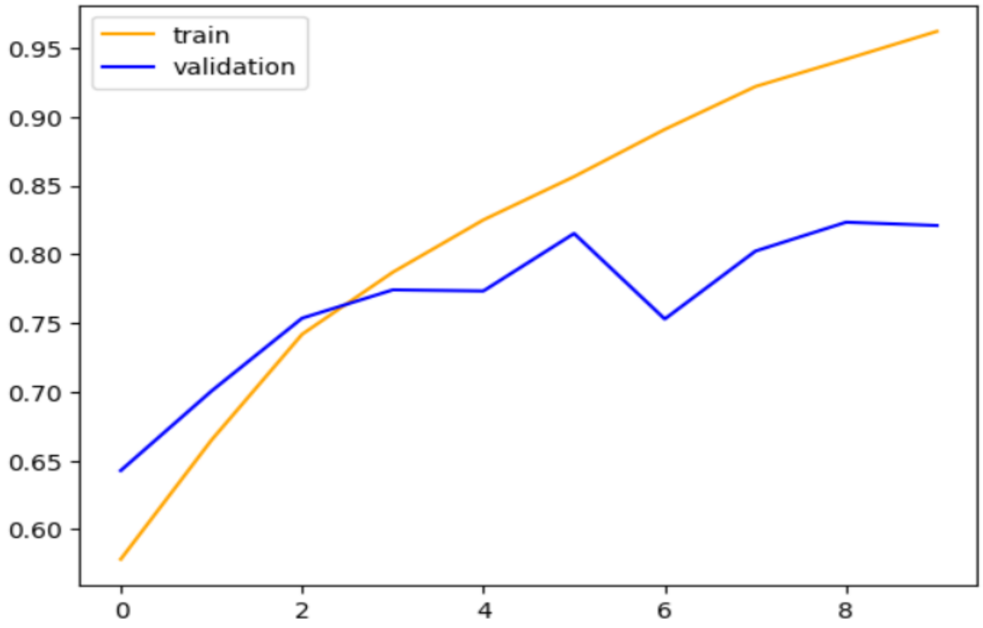
Model2 Summary:
I used convolutional neural networks with 32, 64 and 128 layers with BatchNormalization and Dropout



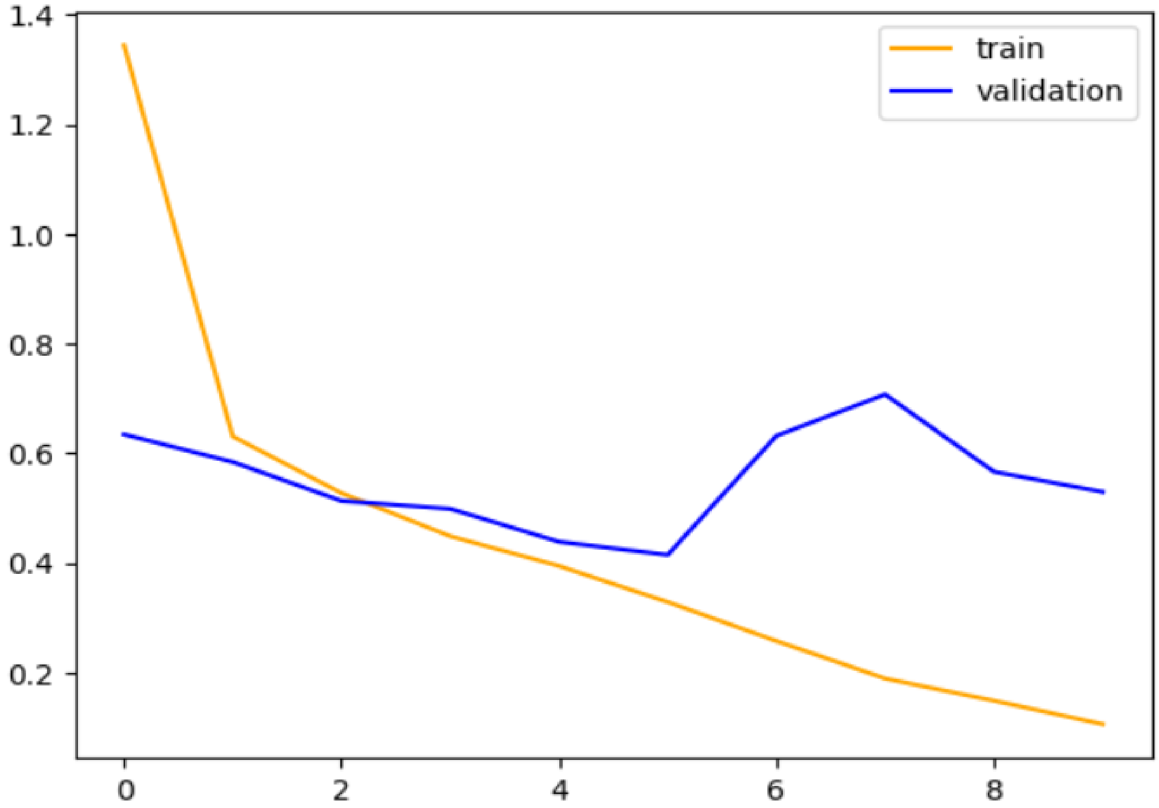
```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d_3 (Conv2D)           (None, 254, 254, 32)      896       
                                                                 
 batch_normalization (Batch  (None, 254, 254, 32)      128       
 Normalization)                                                  
                                                                 
 max_pooling2d_3 (MaxPoolin  (None, 127, 127, 32)      0         
 g2D)                                                            
                                                                 
 conv2d_4 (Conv2D)           (None, 125, 125, 64)      18496     
                                                                 
 batch_normalization_1 (Bat  (None, 125, 125, 64)      256       
 chNormalization)                                                
                                                                 
 max_pooling2d_4 (MaxPoolin  (None, 62, 62, 64)        0         
 g2D)                                                            
                                                                 
 conv2d_5 (Conv2D)           (None, 60, 60, 128)       73856     
                                                                 
 batch_normalization_2 (Bat  (None, 60, 60, 128)       512       
 chNormalization)                                                
                                                                 
 max_pooling2d_5 (MaxPoolin  (None, 30, 30, 128)       0         
 g2D)                                                            
                                                                 
 flatten_1 (Flatten)         (None, 115200)            0         
                                                                 
 dense_3 (Dense)             (None, 128)               14745728  
                                                                 
 dropout (Dropout)           (None, 128)               0         
                                                                 
 dense_4 (Dense)             (None, 64)                8256      
                                                                 
 dropout_1 (Dropout)         (None, 64)                0         
                                                                 
 dense_5 (Dense)             (None, 1)                 65        
                                                                 
=================================================================
Total params: 14848193 (56.64 MB)
Trainable params: 14847745 (56.64 MB)
Non-trainable params: 448 (1.75 KB)
_________________________________________________________________
```

Training and Validation Graph:

Accuracy



Loss



Conclusion : As we can see from both Model’s accuracy plot, Model 2 has performed better as its validation accuracy is increases with each epoch and the gap between Training accuracy and Validation accuracy has reduced.