**Assignment-3**

### Objective: To construct a Convolutional Network on a dataset containing images of cats and dogs so that the model can effectively classify the unseen images (Test data) into cats and dogs by learning the features of these images.

### Data source: The data set was taken from Kaggle, which has Training and Test sets with 25000 and 12500 images, respectively, of the same number of cats and dogs. However, we will only use a few images to build the model (2000).

### 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 you train from scratch. What performance did you achieve?

### *Test Accuracy - 71.9%*

### *Training Accuracy - 98.35%*

### *Validation Accuracy - 77.09%*

### So, we got the above results when we used a training sample of 1000 images and used dropout to reduce overfitting.

### 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?

### *Test Accuracy - 82.70%*

### *Training Accuracy - 88%*

### *Validation Accuracy - 84%*

### As a result, when the number of training samples is increased and data augmentation is utilized as anticipated, there is a significant increase in the accuracy of the Train, Validation, and Test (from 71% to 82.7%).

### 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 the best prediction results.

### Let's increase the Training sample size to 2000 and monitor the results.

### *Test Accuracy - 80%*

### *Training Accuracy - 89%*

### *Validation Accuracy - 81.20%*

### As a result, when the training sample size is increased from 1500 to 2000, the Validation and Test accuracy decreases. Based on the aforementioned observations, 1500 is the ideal training sample size.

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

### Summary

### Using a Pre-trained model without Augmentation

### We reached a validation accuracy of about 97.30% which is better than what we achieved while training the small model from scratch. However, the plots also indicate that we're overfitting almost from the start—despite using a large rate of dropout.

### Using a Pre-trained model without Augmentation

### The set of samples that are used to evaluate a model is always what determines its accuracy! Some example sets might be more troublesome than others, and solid outcomes on one set will just to some extent mean any remaining gatherings. The pre-trained model's accuracy increased slightly from 97.30% to 98.10% with enhancement for this reason.

### Feature extraction with a pre-trained model

### Using the representations learned by a previously trained model—in our case, ImageNet—to extract interesting features from new samples is known as feature extraction. After that, a brand-new, untrained classifier is run through these features.

### A trained VGG16 model with Fine-tuning

### Fine-tuning involves simultaneously training the newly added part of the model—in this case, the fully connected classifier—and a few of the top layers of a frozen model base used for feature extraction. Because it makes the more abstract representations of the reused model slightly more relevant to the issue at hand, this is referred to as "fine-tuning."