

**Summary Report**  
**on**  
**Convolution Assignment**

**Professor**  
**CJ Wu**

**Submitted by**  
**Pavan Chaitanya B**

## **Question:**

Apply convolution networks (convnets) to image data and explain the relationship between sample sizes and the use of training the convnets from scratch versus using a pretrained network.

## **Objective:**

The aim of this project is to use deep learning techniques, specifically convnets, for computer vision in order to accurately classify images. In this report we will explore how these techniques help computers to understand and process visual information, enabling them to perform tasks that humans can do. One example of this is classifying images of cats and dogs using convnets.

There are two main methods for doing this:

- 1) Training a model from scratch.
- 2) Using a Pre-Trained convnet.

When working with smaller datasets, it's important to be mindful of overfitting, which can make prediction more difficult. However, there are techniques available to help overcome overfitting, such as data augmentation, dropouts and we can evaluate the performance on a validation and test set.

## **Data Source:**

The data for this project was obtained from Kaggle and consists of 25,000 training images and 12,500 test images, with an equal number of cats and dogs in each set.

## **Methodology:**

**We have used the large dataset** that is present in Kaggle rather than the small dataset.

Step 1: We have created the Kaggle. json API and uploaded to the Google Colab.

Step 2: We then Copied the Cats VS Dogs data API link

i.e., “**Kaggle competitions download -c dogs-vs-cats**”. And the unzipped the required files that we need.

Step 3: We have built

### **6 models under Scratch Models**

**3 Models under Pre-Trained Models** using various configurations.

These configurations include a different number of layers, different numbers of nodes, optimizers, dropout rates, and other parameters.

### **Analysis:**

#### **Validation Accuracy, Test Accuracy, and Test Loss: (Scratch Models)**

<b>Model no</b>	<b>Training sample size</b>	<b>Validation and Test sample size</b>	<b>Validation Accuracy</b>	<b>Test Accuracy</b>	<b>Test Loss</b>
Model 1	1000	500,500	0.735	0.750	0.529
Model 1a	1000	500,500	0.788	0.830	0.413
Model 1b	1000	500,500	0.748	0.759	0.521
Model 1c	1000	500,500	0.797	0.809	0.434
Model 2	5000	500,500	0.995	0.893	0.891
Model 3	10000	500,500	0.976	0.903	0.285

#### **Validation Accuracy, Test Accuracy, and Test Loss: (Pre-Trained Models)**

<b>Model no</b>	<b>Training Sample Size</b>	<b>Validation and test sample size</b>	<b>Validation Accuracy</b>	<b>Test Accuracy</b>	<b>Test Loss</b>
Model 4	1000	500,500	0.976	0.979	0.191
Model 5	5000	500,500	0.977	0.973	0.131
Model 6	10000	500,500	0.975	0.981	0.071

## **Summary:**

### **Scratch Models:**

1)The Cats and Dogs example using an unregularized model (Model 1) was trained on a sample of 1000, validated on a sample of 500, tested on a sample of 500.

However, the accuracy obtained was only 75%, which indicates overfitting since the training sample was relatively small.

2)We can enhance the model's performance without changing the sample size, which is 1000, by employing various methods. To achieve this, I have used three techniques in the model, namely

Model 1a) Data Augmentation

Model 1b) Dropout Method

Model 1c) Data Augmentation and dropout method.

3) The model's accuracy was observed to increase to 83% when trained using data augmentation technique compared to unregularized model.

4)We tried to improve accuracy by increasing the amount of training samples to 5000(Model 2) and 10000(Model 3), which gave an improved result.

5) We got a validation accuracy of 99% and test Accuracy of 89 % by increasing the training sample size to 5000 (Model 2).

6) By further increasing the training sample size to 10000 (Model 3), the validation accuracy was reduced to 92% and the test accuracy improved to 90%.

**Comparison:** Upon comparison with the unregularized model, it appears that the regularized models have higher validation accuracy and test accuracy.

### **Pre-Trained Models:**

- 1)Pre-Trained Models have better Accuracy and low loss compared with the Scratch Models.
- 2)97 % is the validation Accuracy and test accuracy of Pre-Trained Model with training sizes 1000(Model4).
- 3)Model 5 of Pre-Trained Model have the same accuracy rate as of Model 4.
- 4)With test accuracy of 98% Model 6 has the higher test accuracy compared to Model 4 and Model 5

**Finally, for both Scratch Models and Pre-Trained Models, the training sample size of 10000 gave the highest accuracy.**

### **Conclusion:**

In conclusion, the training sample size is crucial in enhancing the model's accuracy by addressing the issue of overfitting. **The relation between the training sample size and the network choice is that while increasing the sample size can lead to better performance, the selection of the network architecture is also significant.** Simpler network architectures may suffice for smaller sample sizes, while more complex architectures may be required for larger sample sizes to capture the underlying data patterns. Additionally, hyper-tuning parameters such as max-pooling, data augmentation, and dropout method contribute to further improving the model's performance. Pretrained networks are especially beneficial when working with smaller datasets since they are already trained on large datasets and can capture various features.

### **References:**

<https://www.kaggle.com/c/dogs-vs-cats/data>

<https://stackoverflow.com/questions/9031783/hide-all-warnings-in-ipython>

[https://kent.instructure.com/courses/57954/pages/lecture-8-dl-for-computer-vision?module\\_item\\_id=2997918](https://kent.instructure.com/courses/57954/pages/lecture-8-dl-for-computer-vision?module_item_id=2997918)