### **ASSIGNMENT 2 REPORT**

**1. Executive Summary**

Our main goal during this initiative involved developing a model which could differentiate between cat and dog photographs. The cat-versus-dog image classification task provides an intriguing challenge because of animal pose wildness along with multiple background variations and lighting inconsistencies and breed differences. Machine learning approaches normally struggle to accomplish successful generalization in such tasks unless engineers spend substantial time refining features. The evaluation was performed using convolutional neural networks (CNNs) together with transfer learning capabilities from the VGG16 pretrained model at multiple sample training size configurations. A comparison took place between the performance gained by creating new models from scratch and the performance of existing models which pre-learned extensive ImageNet data image representations.

### **2. Project Objectives & Motivation**

The task benefits from deep learning technology since this method allows automatic extraction of hierarchical image features directly from raw data. The spatial ordering of features attributed to CNNs provides superior efficiency when detecting object patterns in addition to its unconventional learning approach. The use of VGG16 as a pretrained model strengthens image detection capability through the implementation of convolutional weights which capture reliable features. The exclusive use of smaller training datasets becomes viable because of the quicker model convergence. Using deep networks in this classification work enables us to dedicate more time to sample scaling and tuning instead of creating features manually.

### **3. Experimental Environment**

The experiment execution occurred through Google Colab as our training environment. The GPU runtime feature from Colab shortens the overall model training duration. The Colab virtual machine utilized standard hardware components which ran on a Tesla T4 GPU. Our main software stack included Python programming while also making use of TensorFlow to train models with Keras APIs and frameworks. The data management and visualization process required additional libraries NumPy, Matplotlib and OS in addition to the main programs. All plots for accuracy and loss tracking were generated using Matplotlib.

The research employed a portion of the original Kaggle Dogs vs. Cats dataset that was organized into train; validation and test subdirectories with cat and dog image kerbs. The Keras image\_dataset\_from\_directory() function utilized the two subfolders cats and dogs in each train validation and test directory. The training process occurred in the train directory while the validation directory assisted model generalization tracking before the performance assessment took place in the test folder. Model behavior testing under low data conditions used the take() method to select specific batch groupings. The model testing procedure became efficient due to the organized structure since it eliminated data duplication while maintaining an easy to navigate setup.

The comprehensive configuration enabled high experimental reliability while making the system workable and allowing both dataset expansion and reduction between evaluations.

### **4. Modeling Approaches**

Two main modeling techniques were tested for image classification purposes in this project: first-hand training of convolutional neural networks (CNNs) alongside transfer learning implementation through the pretrained VGG16 model. A CNN model made from scratch contained multiple sequential convolutional and pooling layers leading to one or more dense layers for processing raw dataset information. Existence of Inception V3 influenced the research because its connection to ImageNet enabled us to implement VGG16 as a static feature encoder combined with customized sequential layers. The method proved helpful for quick training time and improved performance especially under conditions of restricted data availability. Testing took place using multiple training sample sizes with the integrated models.

### **5. Training Scenarios & Sample Sizes (CNN from Scratch)**

The CNN model trained using 1000 images achieved a test accuracy which reached 71.8%. The general improvement in training and validation curves is observed throughout epochs but validation accuracy shows inconsistent behavior that impacts generalization. The model behavior becomes expected because the insufficient sample diversity in the dataset reduces the ability to capture reliable features.

A graph with red and blue lines

AI-generated content may be incorrect.

Using 1500 training samples resulted in a more steady training process that produced test performance of about 65.1%. During training the validation accuracy presented sharp unpredictable movements because of occasional model overfitting behavior. The model learned better training patterns as it stabilized its operation across validation and training accuracy results throughout late epochs.

A graph with red and blue lines

AI-generated content may be incorrect.

The model demonstrated its top generalization capability at 2000 samples where it obtained a test accuracy of 74.4%. The model showed optimal balance between training and validation values which resulted in lowered overfitting effects. The improved forecasting demonstrates that bigger data samples create better results for CNN training when done from scratch.

A graph with red and blue lines

AI-generated content may be incorrect.

The experiments establish a direct connection between the quantity of training data and model accuracy values. Additional data improves generalization yet the analysis shows that returning little extra benefit goes beyond a given limit. Research suggests that adding more than 1500-2000 training samples would not produce significant accuracy improvements unless accompanied by architectural adjustments or optimized parameters.

The VGG16 Transfer Learning process includes training different scenarios while performing inference tasks.

A total of 1000 training samples led the VGG16 model to achieve validation accuracy at 95.8% and test accuracy at 96.4%. The training curve demonstrates accuracy growth stability while the validation accuracy reaches its highest point quickly implying good dataset generalization.

A graph of a graph with a line

AI-generated content may be incorrect.

A higher training sample count to 1500 produced minor increases in both model accuracy when measured for training and validation processing. Both the validation accuracy and test accuracy revealed results of 97.4% and 96.6% respectively. The second set of learning curves demonstrates consistent performance stability because they show a minimal difference between training and validation lines.

A graph of a graph with a line

AI-generated content may be incorrect.

The test accuracy reached 97.4% when 2000 training samples were used which produced the highest measurement. These curves maintain an almost equal distance from one another as they run parallel to each other thus exhibiting perfect model generalization along with minimal overfitting. The VGG16 model demonstrates perfect data utilization when it reaches the accuracy saturation point.

A graph of a graph with a line

AI-generated content may be incorrect.

The experimental results demonstrated that VGG16 surpassed the CNN that started from scratch over every tested sample size. The model achieved validation accuracies higher than 95% with 1000 samples possible through utilizing transfer learning and pretrained weights. The performance improved slightly with additional data input although VGG16 delivered exceptional results with even minimal training data which proves its capability to operate effectively with small and medium-scale datasets.

### **6. Optimization Techniques Applied**

A mix of important optimization methods enabled optimal model generalization and training performance across all experiments. Data augmentation acted as an essential tool to reduce overfitting because it produced artificial data variations which expanded training data variety. Random horizontal flipping combined with rotation and zoom functions provided the model with various object views which enabled it to detect more resilient image patterns.

The fully connected layers within both CNN and VGG16 received vital protection by implementing dropout regularization as a strategy to fight overfitting. During training periods the model lost 0.5 percent of its neurons at random to prevent reliance on specific information thus prompting it to recognize fundamental patterns instead.

The process of varying learning rates directly impacted both training consistency and the convergence rate. VGG16 required a learning rate set to 1e-5 when fine-tuning its pretrained layers in order to update weights gradually during training to prevent significant modifications from the ImageNet learned configuration. Integrity of learned features remained intact through the process that allowed them to adapt to the Cats vs Dogs data.

The training process included early stopping to minimize overfitting and enhance efficiency. The training was stopped by this technique after observing stable validation loss for several consecutive epochs which allowed the maintenance of the optimal model for the task while minimizing the risk of overfitting.

### **7. Performance Insights & Comparisons**

Different model variations and training sample sizes led to important findings through analysis of training and validation patterns. The accuracy of CNN from scratch increased progressively as the training sample size grew. To match the pretrained model performance the required dataset had to increase in size. A sample number of 2000 samples resulted in enhanced test accuracy which reached up to 74%.

VGG16 demonstrated superior performance compared to the scratch CNN at every training sample measurement point. Test accuracy of the pretrained model surpassed 96% even when dealing with only 1000 samples which indicated its ability to perform effectively in low-data condition. The alignment between both curves demonstrated low overfitting effects together with an ideal generalization ability.

### **8. Conclusion**

The analysis reveals VGG16 transfer learning as a substantially superior methodology because it provides better performance while maintaining steady training stability under low-data conditions.

The optimal sample number for transfer learning should be between 1500–2000 samples because additional data samples generate only minimal performance improvements. The performance level of pretrained models reaches maximum accuracy independently of big datasets through appropriate optimization methods.