Summary Report

Introduction:

To categorize photos into two categories—cats and dogs—a variety of deep learning models were developed and assessed for this project. To find the model that best balances accuracy and efficiency, the primary objective was to investigate the effects of various model architectures, activation functions, regularization methodologies, and data augmentation tactics. The goal was to optimize sentiment prediction performance by comparing the performance of more sophisticated and simpler models, with an emphasis on obtaining lower loss and improved classification accuracy.

Data Preparation:

To prepare the dataset, the photos were resized to a standard size and normalized. The diversity of training data was increased by the use of image augmentation, which enhanced the model's capacity for generalization. To validate the model's performance and avoid overfitting, the data was separated into training, validation, and test sets.

Model Development:

A variety of models with various architectures were created, each of which investigated a distinct set of configurations. The number of layers, regularization techniques (like dropout), and activation function types (like Tanh or ReLU) differed among these models. Convolutional, pooling, and fully linked layers were used in the construction of the models. While some models had fewer layers and were simpler, others included more layers for regularization, more convolutional layers, or data augmentation to diversify the training data. The framework of each model was created to address various facets of model performance, such enhancing generalization or decreasing overfitting.

Training:

A batch size of 32 was used to train all models over a predetermined number of epochs using the training dataset. To keep an eye on performance and avoid overfitting, early stopping was employed. Hyperparameters like as learning rates, epoch counts, and dropout rates were adjusted during the training process. Model performance during training and model refinement were assessed using the validation dataset.

Evaluation:

Accuracy and loss were the main performance indicators used to assess each model on the test dataset following training. These criteria were used to compare the efficacy of the various architectures and evaluate how well each model sorted photos into the appropriate categories.

Model Performance:

Each model's performance outcomes on the test dataset are as follows:

Model 1: This baseline model had an accuracy of 64.60% and a loss of 0.9271. The architecture's simplicity resulted in comparatively poor performance.

Layer (type)	Output Shape	Param #
conv2d_13 (Conv2D)	(None, 178, 178, 32)	896
max_pooling2d_13 (MaxPooling2D)	(None, 89, 89, 32)	0
conv2d_14 (Conv2D)	(None, 87, 87, 64)	18,496
max_pooling2d_14 (MaxPooling2D)	(None, 43, 43, 64)	0
conv2d_15 (Conv2D)	(None, 41, 41, 128)	73,856
max_pooling2d_15 (MaxPooling2D)	(None, 20, 20, 128)	0
conv2d_16 (Conv2D)	(None, 18, 18, 128)	147,584
max_pooling2d_16 (MaxPooling2D)	(None, 9, 9, 128)	0
conv2d_17 (Conv2D)	(None, 7, 7, 256)	295,168
max_pooling2d_17 (MaxPooling2D)	(None, 3, 3, 256)	0
flatten_3 (Flatten)	(None, 2304)	0
dense_6 (Dense)	(None, 512)	1,180,160
dropout_2 (Dropout)	(None, 512)	0
dense_7 (Dense)	(None, 1)	513

Model 2: This model greatly improved with extra convolutional layers and filters, attaining an accuracy of 78.8% and a loss of 0.4629. This demonstrated the advantages of giving the network more depth.

Layer (type)	Output Shape	Param #
input_layer_6 (InputLayer)	(None, 180, 180, 3)	0
sequential_5 (Sequential)	(None, 180, 180, 3)	0
rescaling (Rescaling)	(None, 180, 180, 3)	0
conv2d_18 (Conv2D)	(None, 178, 178, 32)	896
max_pooling2d_18 (MaxPooling2D)	(None, 89, 89, 32)	0
conv2d_19 (Conv2D)	(None, 87, 87, 64)	18,496
max_pooling2d_19 (MaxPooling2D)	(None, 43, 43, 64)	0
conv2d_20 (Conv2D)	(None, 41, 41, 128)	73,856
max_pooling2d_20 (MaxPooling2D)	(None, 20, 20, 128)	0
conv2d_21 (Conv2D)	(None, 18, 18, 256)	295,168
max_pooling2d_21 (MaxPooling2D)	(None, 9, 9, 256)	0
conv2d_22 (Conv2D)	(None, 7, 7, 256)	590,080
max_pooling2d_22 (MaxPooling2D)	(None, 3, 3, 256)	0
flatten_4 (Flatten)	(None, 2304)	0

Model 3: By adding dropout regularization, this model performed even better, attaining an accuracy of 80.5% and a loss of 0.451. Dropout was included to enhance generality and avoid overfitting.

Model 4: With a loss of 0.6851 and an accuracy of 54.2%, this more straightforward model did not perform well, indicating that a basic architecture devoid of regularization is not enough to achieve decent results.

Model 5: This model outperformed the simpler models but still fell short of the deeper architectures, with an accuracy of 71.2% and a loss of 0.5565 with a mix of additional filters and dropout.

Model 6: A model with a loss of 0.484 and an accuracy of 79.0% was produced by utilizing strides and max-pooling. Through the extraction of more pertinent features from the data, this model's performance increased.

Model 7: This model obtained an accuracy of 75.18% and a loss of 0.5397 by including data augmentation and dropout.

Model 8: This model, which combined strides, max-pooling, and data augmentation, demonstrated that these tactics significantly improved classification performance with an accuracy of 80.4% and a loss of 0.441.

Model 9: This model likewise achieved an accuracy of 80.4% and a loss of 0.441 with sophisticated regularization and optimizations. Its performance was in line with Model 8, indicating that stable outcomes can be achieved by regularization and fine-tuning.

In summary:

The models that included data augmentation and advanced regularization approaches, including dropout, performed the best and had the highest accuracy (80.4%).

In general, models with more layers and intricate structures outperformed simpler models. Complexity does not always translate into better outcomes, as evidenced by the fact that certain models with fewer layers performed well and that adding more layers did not always result in a noticeable improvement.

In order to attain good performance, more complex architectures and regularization are required, as demonstrated by the poor performance of the baseline model and the more basic models (Models 4 and 5).

Ideas:

Use models with sophisticated regularization methods (such as dropout) and data augmentation to get the best results.

To strike a compromise between regularization and information retention, think about optimizing dropout rates.

Steer clear of superfluous layers if they don't increase the accuracy of the model. Rather, concentrate on improving the current regularization methods and layers.

Model performance can also be improved by fine-tuning hyperparameters like learning rate, batch size, and epoch count.