This project involved designing a convolution network (convnet) to maximize performance on an image classification model. In this model, we are classifying images of cats and dogs as cats or dogs. We are using validation and test sets of 500 images of cats and 500 images of dogs, with varying sizes of training sets. The training set size is for cats and dogs each, so a training set of 500 would be 500 images of cats and 500 images of dogs. Maximizing the performance consists of training set size adjustments, as well as incorporating data augmentation and dropout to reduce the risk of overfitting. Additionally, we will be leveraging a pretrained model with layer freezing as well. Model performance is being determined by test set accuracy, and the model is being designed in Python.

First, we trained a convnet from scratch (as opposed to using a pretrained network). We trained models with data augmentation + dropout as well as without. The results are below. The test accuracy percentage can be determined by multiplying the result times 100.

Model created from scratch		
Training set size	No augmentation or dropout	With augmentation + dropout
500	0.667	0.706
750	0.697	0.772
1000	0.704	0.837

Then, we leveraged a pretrained network, using the VGG16 architecture for image classification. For the pretrained network, we also incorporated layer freezing- allowing the last 4 layers of the model to be fine-tuned through training.

Pretrained network			
Training set size	Dropout only	Augmentation + dropout	Augmentation + dropout + layer freezing
300	0.961	0.956	0.961
500	0.963	**0.976**	0.972
600	0.960	0.969	0.968
750	0.971	0.973	0.974
1000	0.970	**0.980**	0.969

Overall, leveraging a pretrained network was significantly more effective than training a network from scratch. Even the best performing model trained from scratch was less effective than the worst performing pretrained model. Regarding training a model from scratch, it appears that increasing training size is positively correlated with performance. Incorporating data augmentation and dropout increased performance for all 3 training sizes. As for the pretrained network, it appears the training sets of 750 and 1000 performed well across the board, though a training set of 500 with data augmentation + dropout also performed quite well, with a 97.6% accuracy. Incorporating layer freezing alongside data augmentation and dropout did not have a consistent improvement to test accuracy, with the best performances coming from data augmentation + dropout. The best performing model at 98.0% accuracy was a pretrained model with a training set of 1000, incorporating data augmentation + dropout. The second best at 97.6% was a similar model with a training set of 500. This suggests that while the largest training set produced the best result, very comparable results can also be achieved with less resources. Additionally, it suggests that layer freezing may not always be worth the added complexity.