# Images Classification of Dogs and Cats using Fine-Tuned VGG Models

Mahardi[1], I-Hung Wang[2], Kuang-Chyi Lee[3*], Shinn-Liang Chang[4]

[1,3]Department of Automation Engineering, National Formosa University
[2,4]Department of Power Mechanical Engineering, National Formosa University, Yunlin County, Taiwan
[*]Corresponding Author: Email: kclee@gs.nfu.edu.tw

## Abstract

Image classification has become more popular as it is the most basic application and implementation of deep learning. Images of dogs and cats are the most common example to train image classifiers as they are relatable. It is easy to classify the image of cats and dogs, but the images of various breeds are difficult to classify with high accuracy. In this paper, we tried to build an image classifier to recognize various breeds of dogs and cats (CDC) using fine-tuned VGG models. Two common models, VGG16 and VGG19 were used to build the classifier. The resulting model from VGG16 has a training accuracy of 98.47%, validation accuracy of 98.56%, and testing accuracy of 83.68%. The model from VGG19 has a training accuracy of 98.59%, validation accuracy of 98.56%, and testing accuracy of 84.07%.

**Keywords:** image classification, deep learning, Keras, VGG

## Introduction

In recent years, deep learning has become popular to solve many problems. Peterson and Gibson [1] mentioned in their book that deep learning used a computational model that shares some properties of how the human brain works. It is built from several neural network layers that are given weight and can be trained by a learning algorithm to minimize the error of the output. The deep network for the image classification is called Convolutional Neural Network (CNN). Buduma and Lacascio [2] mentioned that CNN used layers of convolutional filters to arrange the network into three dimensions: width, height, and depth. The depth of the layers allows the filter to combine the information from all the learned features. The problem is that training data for CNN is not easy to create, as CNN requires a large amount of training data and training is time-consuming. The alternative to train CNN is a fine-tuned CNN model that is trained from another application. Tajbakhsh *et al*. [3] proved that using fine-tuned CNNs outperformed fully trained CNNs especially when there was limited training data. Manaswi [4] mentioned that fine-tuning replaces and retrains the classifier on the top of CNN and fine-tune the weight of the pre-trained network via backpropagation. Image classification of dogs and cats is a common problem that practices deep learning. In 2012, Parkhi *et al.* [5] proposed the method to classify the images of 37 different breeds of dogs and cats with an accuracy of 59%. In 2018, Panigrahi *et al.* [6] used deep learning to classify images of dogs and cats with an accuracy of 88.31%. The objective of this study is to develop a better image classifier for various breeds of dogs and cats, later called CDC by a fine tuning VGG model.

## Dataset for CDC

The pre-trained models are developed for general image classification. To classify the new classes, the models have to be trained with the new dataset using the labeled new classes. In this paper, pictures of 21 different breeds of dogs and cats were used as the dataset to be trained for a CDC model which were separated into three datasets: training data, validation data, and test data.

TABLE 1
PICTURE QUANTITY OF VARIOUS BREEDS OF DOGS AND CATS

| Label No. | Category | Breed | Training data | Validation data | Test Data |
|---|---|---|---|---|---|
| 0 | Dog | Akita | 818 | 102 | 103 |
| 1 | Dog | Alaskan Malamute | 992 | 124 | 124 |
| 2 | Dog | Basenji | 918 | 114 | 116 |
| 3 | Dog | Basset Hound | 967 | 120 | 122 |
| 4 | Dog | Beagle | 956 | 119 | 120 |
| 5 | Dog | Belgian Malinois | 662 | 82 | 84 |
| 6 | Dog | Bernese Mountain Dog | 1233 | 154 | 155 |
| 7 | Dog | Border Collie | 1224 | 153 | 153 |
| 8 | Dog | Boston Terrier | 960 | 120 | 120 |
| 9 | Cat | Norwegian Forest Cat | 976 | 122 | 123 |
| 10 | Dog | Shiba Inu | 1293 | 163 | 161 |
| 11 | Cat | Abyssinian Cat | 1045 | 132 | 131 |
| 12 | Cat | American Short Hair | 1233 | 154 | 155 |
| 13 | Cat | Birman Cat | 896 | 112 | 113 |
| 14 | Cat | Cornish Rex | 757 | 94 | 96 |
| 15 | Cat | Devon Rex | 746 | 93 | 94 |
| 16 | Cat | Maine Coon | 1134 | 141 | 143 |
| 17 | Cat | Scottish Fold | 1128 | 141 | 142 |
| 18 | Cat | Siamese Cat | 718 | 90 | 91 |
| 19 | Cat | Siberian Cat | 1186 | 149 | 151 |
| 20 | Cat | Somali Cat | 732 | 93 | 93 |
| | **Total** | | **20574** | **2572** | **2590** |

Each of the main folders has a subfolder that separates the breed of dogs and cats. The quantity of each breed is shown in Table 1. The source of the images is the Dreamtime Stock Photos [7]. The total number of pictures in this experiment is 20,574 for training data, 2,572 for validation data, and 2,590 for test data.



*ISBN: 978-1-7281-8060-1*

### Fine Tuning Models for CDC

In 2014, Simoyan *et al.* [8] developed a model of Convolutional Network called VGG. It consists of layers of 3 x 3 convolutional filter which has a small receptive field. The convolution filters are separated into five blocks with a max-pooling filter as spatial pooling at the end of each block. The two most commonly used models with VGG configuration are VGG16 and VGG19. VGG16 consists of 13 convolutional layers and 3 fully connected layers, while VGG19 has 16 convolutional layers and 3 fully connected layers. The configuration of VGG models is shown in Table 2. The width of the layers is small, starting from 64 in the first layer and increasing by the factor of 2 in each block until it reaches 512. The input of VGG has a fixed size 224 x 224 RGB image.

TABLE 2
CONFIGURATION OF VGG MODELS

| ConvNet Configuration | |
|---|---|
| 16 weight layers | 19 weight layers |
| input (224 × 224 RGB image) | |
| conv3-64<br>conv3-64 | conv3-64<br>conv3-64 |
| maxpool | |
| conv3-128<br>conv3-128 | conv3-128<br>conv3-128 |
| maxpool | |
| conv3-256<br>conv3-256<br>**conv3-256** | conv3-256<br>conv3-256<br>conv3-256<br>**conv3-256** |
| maxpool | |
| conv3-512<br>conv3-512<br>**conv3-512** | conv3-512<br>conv3-512<br>conv3-512<br>**conv3-512** |
| maxpool | |
| conv3-512<br>conv3-512<br>**conv3-512** | conv3-512<br>conv3-512<br>conv3-512<br>**conv3-512** |
| maxpool | |
| FC-4096 | |
| FC-4096 | |
| FC-1000 | |
| softmax | |

To train the models into a new set of classes, we took out the top layer of the pre-trained VGG model and redefined the fully connected layers into new classes. After the model was built, the model was retrained with the training and validation dataset. The parameters are shown in Table 3. We used 50 epochs with a batch size of 16. Optimizer Stochastic gradient was used with a learning rate of 0.0001 and binary cross-entropy as the loss function.

TABLE 3
PARAMETER FOR TRAINING THE MODELS

| Model | Epochs | Batch Size | Learning Rate | Loss Function | Optimizer |
|---|---|---|---|---|---|
| VGG16 | 50 | 16 | 0.0001 | binary crossentropy | SGD |
| VGG19 | 50 | 16 | 0.0001 | binary crossentropy | SGD |

### CDC Models Training Result

The training result using the VGG16 model is shown in Fig. 1, which has two graphs of how much the accuracy and loss function by each epoch of training. While doing the training, the validation data is used to check the validation accuracy so that the model is not overfitting into the training data. The accuracy of the training data of CDC based on VGG16 is 98.47% and the validation accuracy in this training is 98.56%. The time consumed for training CDC by the VGG16 model is 6 hours and 36 minutes.

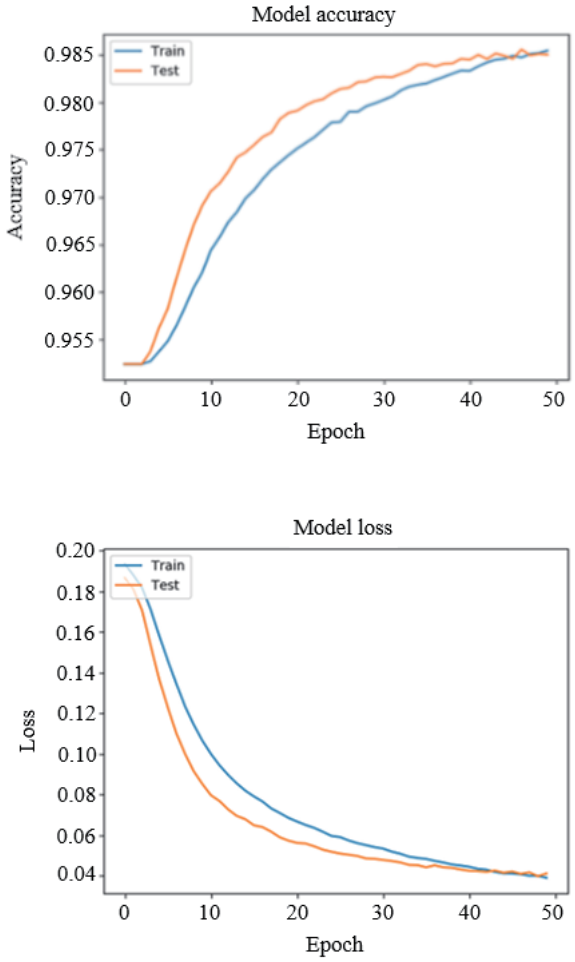


Fig. 1 Graph of accuracy and loss by each epoch for training using VGG16.

The graph of accuracy and loss in training using VGG19 is shown in Fig. 2. The training time for VGG19 based CDC is 6 hours and 38 minutes. The training accuracy is 98.59% and the validation accuracy in this training is 98.56%. The comparison of the accuracy of both models is shown in Table 4.

ISBN: 978-1-7281-8060-1

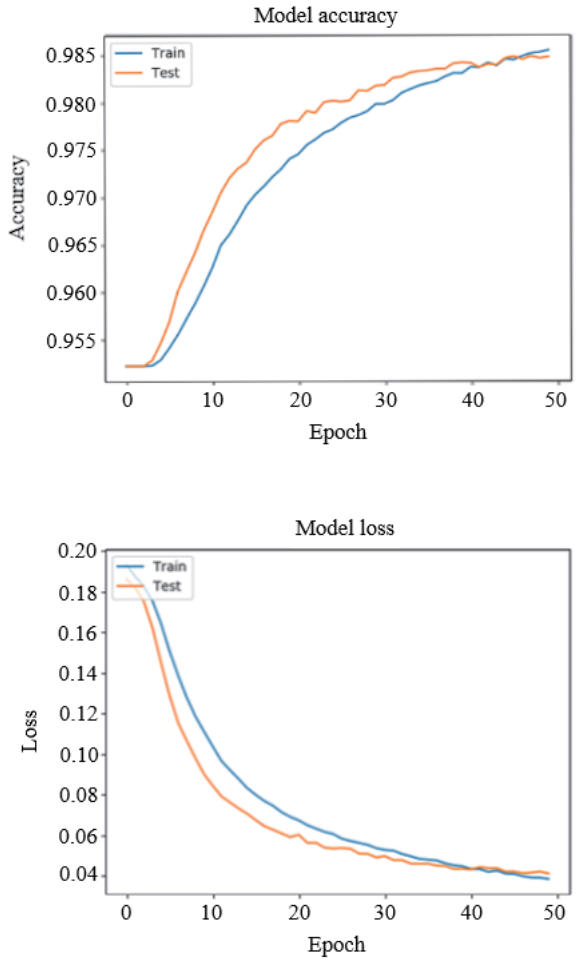


Fig. 2 Graph of accuracy and loss by each epoch for training using VGG19.

TABLE 4
TRAINING RESULT OF VGG MODELS

| Model | Loss | Accuracy | Validation Loss | Validation Accuracy |
|---|---|---|---|---|
| VGG16 | 4.07% | 98.47% | 4.09% | 98.56% |
| VGG19 | 3.77% | 98.59% | 4.02% | 98.56% |

After the training was finished, the testing accuracy of CDC was calculated using a confusion matrix that was created from the prediction using the test data folder of a dataset. The result of generating confusion matrix of VGG16 based CDC is shown in Table 5 which shows the prediction and recall the percentage of each class. By Eq. (1), the testing accuracy of VGG16 based CDC is 83.68%.

$$\text{Accuracy}(\%) = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + FN_i} \times 100\% \qquad (1)$$

where $TP_i$ is the true positive value for each class, $FN_i$ is the false negative value for each class, and n is 21 classes of dogs and cats.

TABLE 5
CONFUSION MATRIX OF TEST DATASET USING VGG16

| pred | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | Rec |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| label | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 71 | 5 | 2 | 0 | 0 | 2 | 0 | 1 | 1 | 0 | 19 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 69% |
| 1 | 3 | 117 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 94% |
| 2 | 0 | 0 | 113 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 97% |
| 3 | 0 | 0 | 2 | 111 | 7 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 91% |
| 4 | 0 | 1 | 2 | 10 | 106 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 88% |
| 5 | 0 | 0 | 0 | 1 | 1 | 75 | 0 | 0 | 0 | 0 | 4 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 89% |
| 6 | 0 | 0 | 0 | 1 | 0 | 1 | 145 | 4 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 94% |
| 7 | 2 | 1 | 0 | 1 | 1 | 1 | 5 | 136 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 89% |
| 8 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 2 | 115 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 96% |
| 9 | 0 | 1 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 78 | 1 | 0 | 0 | 4 | 0 | 0 | 9 | 3 | 1 | 21 | 1 | 63% |
| 10 | 10 | 2 | 5 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 136 | 2 | 0 | 0 | 1 | 1 | 0 | 2 | 0 | 0 | 1 | 84% |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 128 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 3 | 96% |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 147 | 0 | 0 | 3 | 1 | 1 | 0 | 2 | 0 | 95% |
| 13 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 1 | 4 | 0 | 77 | 3 | 0 | 1 | 2 | 8 | 8 | 2 | 68% |
| 14 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 3 | 0 | 0 | 71 | 15 | 0 | 1 | 2 | 1 | 0 | 74% |
| 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 4 | 1 | 1 | 12 | 70 | 0 | 2 | 0 | 1 | 1 | 74% |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 19 | 1 | 0 | 0 | 2 | 0 | 0 | 96 | 0 | 0 | 18 | 7 | 67% |
| 17 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 1 | 2 | 1 | 0 | 126 | 0 | 5 | 2 | 89% |
| 18 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 1 | 6 | 0 | 0 | 1 | 0 | 0 | 0 | 79 | 0 | 1 | 87% |
| 19 | 0 | 3 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 20 | 1 | 0 | 3 | 7 | 0 | 0 | 7 | 3 | 1 | 100 | 4 | 66% |
| 20 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 6 | 0 | 9 | 1 | 0 | 0 | 0 | 1 | 0 | 2 | 1 | 72 | 77% |

The confusion matrix was also generated for CDC based on the VGG19 model that was fine-tuned and finished training using the training and validation dataset. The confusion matrix of VGG19 based CDC is shown in Table 6 with the testing accuracy at 84.07%.

TABLE 6
CONFUSION MATRIX OF TEST DATASET USING VGG19

| pred | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | Rec |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| label | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 73 | 2 | 3 | 0 | 0 | 3 | 0 | 3 | 2 | 0 | 16 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 71% |
| 1 | 2 | 113 | 0 | 0 | 0 | 0 | 1 | 3 | 0 | 1 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 91% |
| 2 | 0 | 0 | 114 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 98% |
| 3 | 0 | 1 | 1 | 109 | 8 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 89% |
| 4 | 1 | 0 | 0 | 5 | 111 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 93% |
| 5 | 0 | 0 | 3 | 2 | 0 | 73 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 87% |
| 6 | 0 | 0 | 0 | 0 | 1 | 0 | 149 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 96% |
| 7 | 0 | 0 | 0 | 1 | 0 | 1 | 5 | 142 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 93% |
| 8 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 2 | 115 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 96% |
| 9 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 84 | 0 | 1 | 1 | 3 | 0 | 0 | 10 | 1 | 3 | 15 | 2 | 68% |
| 10 | 12 | 2 | 3 | 0 | 1 | 3 | 0 | 0 | 0 | 0 | 134 | 2 | 0 | 0 | 1 | 0 | 0 | 2 | 1 | 0 | 0 | 83% |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 127 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 95% |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 148 | 0 | 0 | 1 | 0 | 2 | 0 | 1 | 0 | 95% |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 2 | 1 | 77 | 1 | 1 | 2 | 5 | 10 | 8 | 2 | 68% |
| 14 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 5 | 0 | 0 | 75 | 10 | 1 | 0 | 1 | 1 | 0 | 78% |
| 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 6 | 2 | 0 | 12 | 67 | 0 | 2 | 1 | 0 | 1 | 71% |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 0 | 0 | 1 | 4 | 0 | 0 | 108 | 0 | 0 | 7 | 8 | 76% |
| 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | 4 | 2 | 1 | 0 | 1 | 125 | 0 | 6 | 0 | 88% |
| 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 4 | 1 | 1 | 0 | 0 | 81 | 0 | 1 | 89% |
| 19 | 0 | 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 21 | 0 | 1 | 2 | 12 | 1 | 0 | 12 | 5 | 0 | 89 | 4 | 59% |
| 20 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 1 | 15 | 1 | 0 | 0 | 0 | 2 | 0 | 1 | 3 | 65 | 70% |

From the testing, VGG19 based CDC has a slightly higher accuracy than VGG16 based CDC, but VGG16 has a higher recall rate in 14 classes than VGG19. VGG16 has a prediction accuracy of 97% for class Basenji and 63% for class Norwegian Forest Cat, while VGG19 has a prediction accuracy of 98% for class Basenji and 59% for class Siberian Cat. The error for class Norwegian Forest Cat is caused as Siberian Cat is predicted as Norwegian Forest Cat. The comparison of Norweigian Forest Cat and Siberian Cat is shown in Fig. 3.

(a) Norwegian Forest Cat (b) Siberian Cat

Fig. 3 Comparison of Norwegian Forest Cat And Siberian Cat.



ISBN: 978-1-7281-8060-1

Furthermore, we use the models to predict individual images of random cats and dogs to check the speed of predicting images. The results show the correct prediction of all pictures. The prediction images are shown in Fig. 4 and the speed of the prediction is shown in Table 7.


Fig. 4 Prediction result of single image

TABLE 7
SPEED OF CDC PREDICTION

| Prediction Time (s) | VGG16 | VGG19 |
|---|---|---|
| Benese Mountain Dog | 4.16 | 4.58 |
| Abyssinian Cat | 4.05 | 4.43 |
| Basset Hound | 4.18 | 4.65 |
| Scottish Fold | 4.40 | 4.75 |
| Beagle | 4.09 | 4.54 |
| **Average** | **4.18** | **4.59** |

The sample test result shows that the prediction speed of both models is around 4 seconds. VGG19 based CDC spends 0.5 second longer time than VGG16 based CDC.

## Conclusion

In this experiment, we developed an image classifier for classifying breeds of dogs and cats (CDC). We retrain VGG16 and VGG19 with new classes to classify the images of various breeds of dogs and cats. Both models have the same validation accuracy of 98.56%, but VGG19 has a better training accuracy of 98.59% and testing accuracy of 84.07% than VGG16 with a training accuracy of 98.47% and testing accuracy of 83.68%. The lowest recall for VGG19 is 59% for class Siberian Cat and that for VGG16 is 63% for class Norwegian Forest Cat. The prediction speed of CDC of individual images has an average prediction time of 4.59 seconds for VGG19 based CDC and 4.18 seconds for VGG16 based CDC.

## References

[1] J. Patterson and A. Gibson, Deep Learning. Sebastopol, U.S. State: O'Reilly Media, Inc., 2017.
[2] N. Buduma and N. Lacascio, Fundamentals of Deep Learning. Sebastopol, U.S. State: O'Reilly Media, Inc., 2017.
[3] N. Tajbakhsh et al., "Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?," in *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1299-1312, May 2016.
[4] N. K. Manaswi, Deep Learning with Applications Using Python. Banglore, India: Apress, 2018.
[5] Parkhi et al. "Cats and dogs," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 3498-3505.
[6] Panigrahi et al., "Deep Learning Approach for Image Classification," 2018 2nd International Conference on Data Science and Business Analytics (ICDSBA), Changsha, 2018, pp. 511-516.
[7] "Dreamtime Stock Photos," [Online] Available: https://www.dreamstime.com/photos-images/dreamtime.html
[8] Simonyan, Karen, et al. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556*. 2014.

*ISBN: 978-1-7281-8060-1*