Dog Breed Identification: An application of Convolutional Neural Network and Transfer Learning on Inception and EfficientNet

Jaki Tang
Master of Science in Analytics
University of Chicago
jakitang@uchicago.edu

Yikai Chen
Master of Project Management
Northwestern University
halfnoir@gmail.com

Abstract

Dogs have lots of breeds. Some of the breeds are very similar to each other and most of the people cannot distinguish. In this project, we want to investigate how current image classification neural networks perform on classifying dog breeds. We plan to start from applying either baseline CNN or pre-trained models like Efficient Net B0 and Inception V3. Then, we optimize the models through adding layers like drop out or batch normalization and through hyper parameter tuning of the optimizers. Without image preprocessing, we achieved train accuracy 96.91% and validation accuracy 42.42% for Inception, and train accuracy 99.94% and validation accuracy 64.25% for Efficient Net. After image preprocessing, the validation accuracy increases to 78.96% for Inception and 79.49% for Efficient Net.

Table of Contents

1 Exploratory Data Analysis

Our dataset is a subset of ImageNet [1] containing 10222 images for training and 10000 images for testing.

1.1 Data Count and Description

The dataset contains 10222 images of 120 dog breeds in train dataset with a csv file containing each image's label information. Average images per dog breed is 85 images. The dataset also contains 10000 unlabeled images as test dataset. There are no missing values in the dataset.

1.2 Visualization of the Raw Data

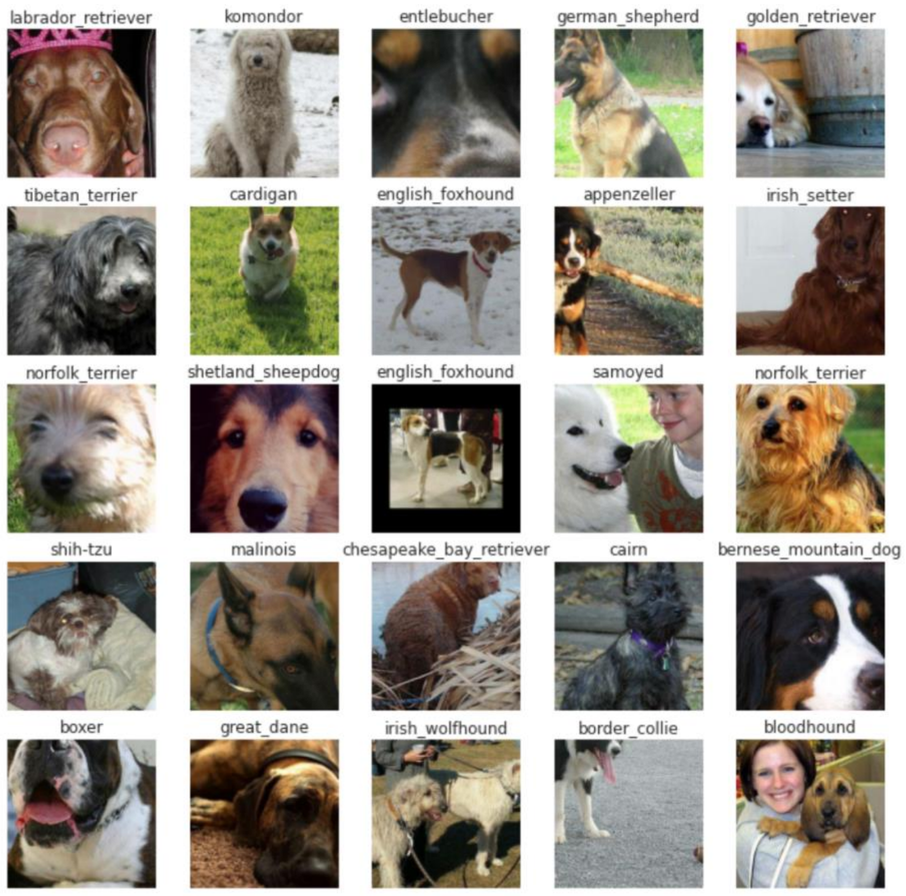


Figure 1: Raw Data Visualization

Figure 1 is a visualization showing some of the training dataset's images. Each image of dog is labeled with its corresponding breed.

1.3 Univariate Analysis

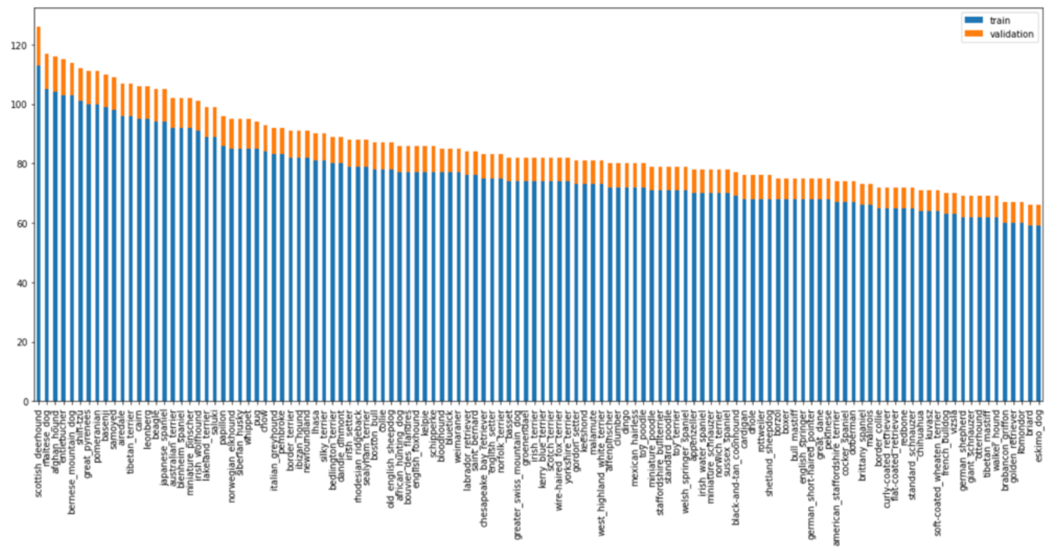


Figure 2: Train and Test Data Distribution

Above is the visualization of a bar chart showing the distribution of images per dog breed in the training dataset. From the univariate analysis, the maximum images in one dog breed are 126 for scottish_deerhound and the minimum images in one dog breed is 66 for eskimo_dog or briard. The standard deviation is 13.24. Therefore, the train dataset is in a good distribution without selection bias.

2 Model Building

We first implemented the baseline CNN and make a test run on our dataset to decide the direction of our next step.

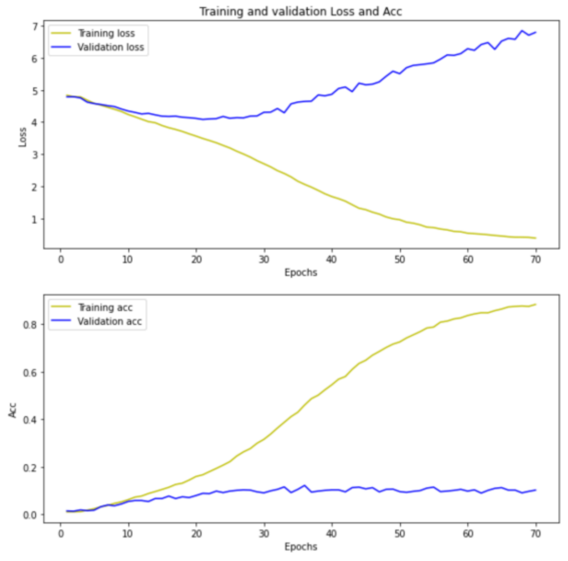


Figure 3: Train and Validation Loss for Baseline CNN

The Figure 3 shows that baseline CNN cannot handle the images’ complexity in our dataset.

2.1 Deep Learning Algorithm Description

Against the baseline CNN. We noticed that the baseline CNN cannot capture all the features in our dataset. For example, the lack of width and depth coefficient in baseline CNN fails to extract necessary features from the images. Therefore, we choose to build one Efficient Net B0 and one Inception V3. Both Inception and Efficient Net are deeper and wider than Baseline CNN thus should give us better results.

2.1.1 Inception V3

We choose InceptionV3. Inception marks an important milestone in the development of CNNs. Inception adopts Inception module which contains multiple kernels with different sizes so that objects of various sizes can be captured from the previous layer. This feature is important to our dataset as dog in each image has different size so a fixed size kernel cannot handle all our data properly. We choose V3 over V1 because V3 Inception factorized those 3x3 and 5x5 kernels so that the computational efficiency is increased [2].

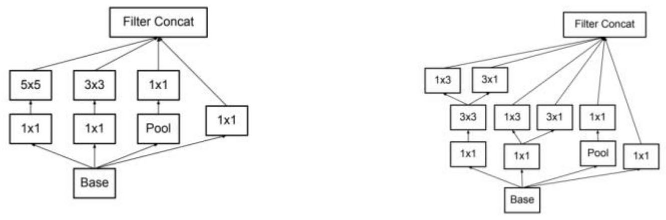
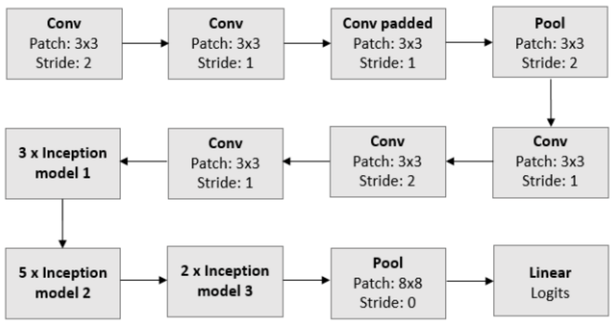


Figure 4: How Inception Module Works for Multiple Kernel Sizes



Conv Patch: 3x3 Stride: 2
Conv Patch: 3x3 Stride: 1
Conv padded Patch: 3x3 Stride: 1
Pool Patch: 3x3 Stride: 2
Conv Patch: 3x3 Stride: 1
Conv Patch: 3x3 Stride: 2
Conv Patch: 3x3 Stride: 1
3 x Inception model 1
5 x Inception model 2
2 x Inception model 3
Pool Patch: 8x8 Stride: 0
Linear Logits

Figure 5: Inception V3 Design

In addition, Inception V3 adds more regularization to the loss function which could prevent overfitting. Because many dog breeds are very similar, we don't want our model to be too confident in classification.

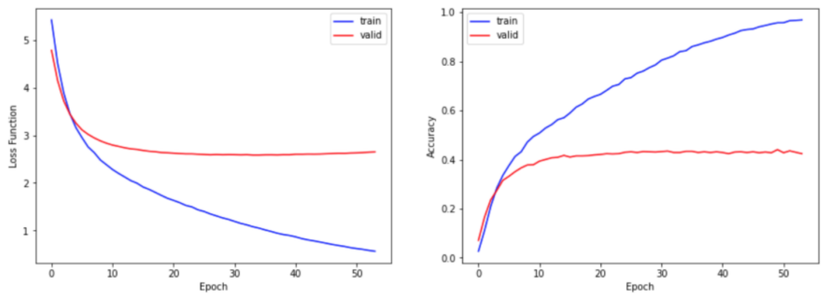


Figure 6: Train and Validation Loss for Inception before Preprocessing

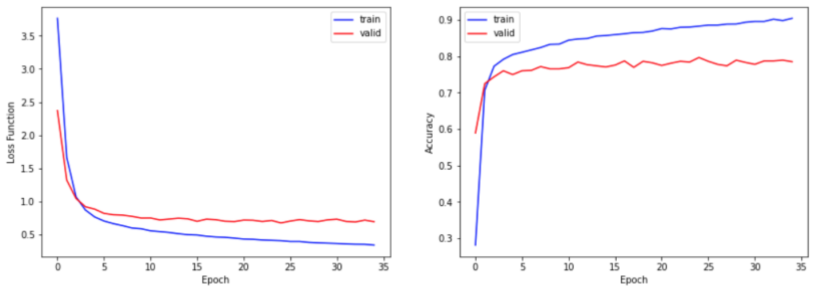


Figure 7: Train and Validation Loss for Inception after Preprocessing

The Figure 5 and 6 shows that our inception model encounter overfitting in the beginning and it cannot be solved simply through model hyper parameter tuning. Therefore, we add normalization to rescale images' pixels by 255 in the preprocessing step. Then, the overfitting issue is much less severe as the validation accuracy increases from 42.42% to 78.96%.

2.1.2 Efficient Net B0

We choose EfficientNet because it is pretrained on more than a million images from the ImageNet database. EfficientNet is newly developed in 2019. It refines the scaling of all dimensions of the neural network like width, depth, and image resolution against the current CNNs to improve the overall performance [6]. The technique used by EfficientNet is compound coefficient. The compound scaling combines the design of width scaling, depth scaling and higher resolution scaling in one design with fixed coefficients. The three coefficients control widths, depths and resolution, and will be balanced with a constant ratio.

The normal grid search achieves total FLOPs (Floating Point Operations Per Second) around 2 but EfficientNet increases this by 2 of the computational resources' square [5].

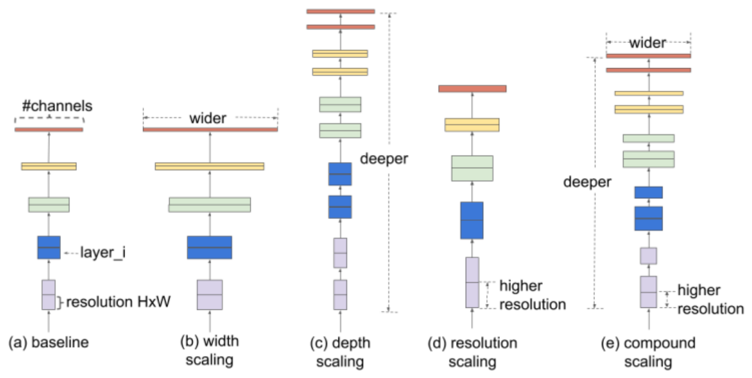


Figure 8: Scaling Method of Efficient Net

The design of EfficientNet is developed by performing a neural architecture search using the AutoML MNAS framework. It adds the FLOPs increase layers to the original design and it balances accuracy and time complexity by penalizing heavy computations because of the large scaling process [6].

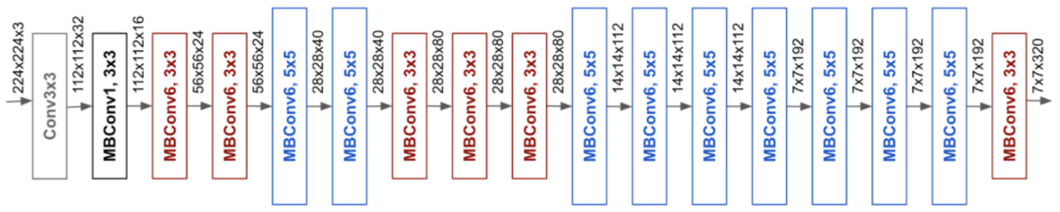


Figure 9: Design of Efficient Net

We choose B0 as our version because the accuracy of B0 is closer to InceptionV3 and we want to perform some model tuning on our own

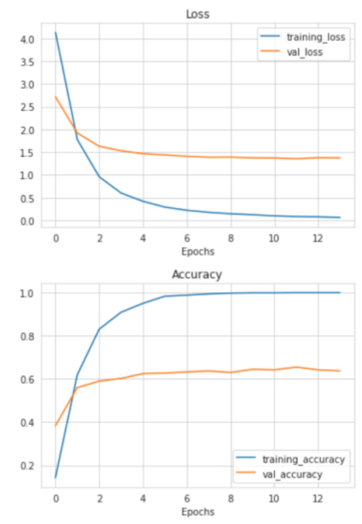


Figure 10: Train and Validation Loss for Efficient Net before Preprocessing

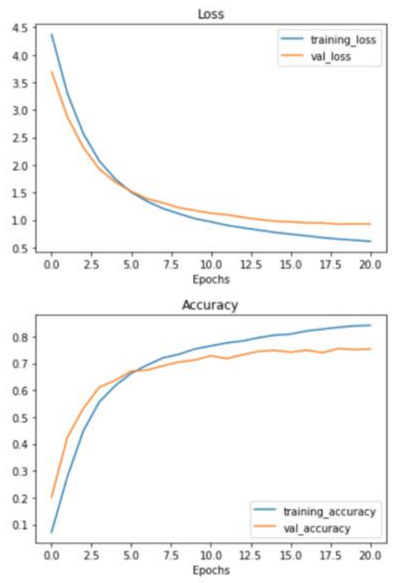


Figure 11: Train and Validation Loss for Efficient Net after Preprocessing

The Figure 10 and 11 shows that our Efficient Net model encounter overfitting in the beginning. However, the overfitting is not as worse as Inception. We think it may be the reason that the Efficient Net is pretrained on ImageNet. Therefore, the overfitting is not as worse as Inception. In addition, we add normalization to rescale images' pixels by 255 in the preprocessing step. Then, the overfitting issue is much less severe as the validation accuracy increases from 64.25% to 79.49%.

2.2 Model Performance Evaluation and Metrics Selection

The model performance evaluation is done through categorical_crossentropy and accuracy.

2.2.1 Loss Function

We use categorical_crossentropy over binary_crossentropy because our encoded output label has 120 classes. And the output label is assigned one-hot category encoding value in form of 0s and 1s [3]. The binary crossentropy would fail to capture the losses between our 120 labeled classes.

2.2.2 Accuracy Metric

We use accuracy as our metrics. If we choose to use binary accuracy, the 120 classes of our label would result in high binary accuracy for all our predictions. Accuracy can better reflect our performance.

Above Figures (Figure 5, 6, 10, and 11) show how our models perform. The results shown some overfitting even when we add dropout layers, batch normalization, and tune the optimizer with learning rate and decay.

Therefore, we choose to preprocess our image with rescale equal to 1.0/255.0 so that all the images are normalized. The results of models processing normalized images are shown below. The fixes some of the overfitting.

3 Model Management

3.1 Model Architecture Deployment

3.1.1 Inception

Based on the Inception, we also add one global_average_pooling layer, two dropout layers, two batch normalization layers, and two dense layers.

3.1.2 Efficient Net

Based on the Efficient Net, we add one global_average_pooling layer as well as one dense layer. The global_average_pooling layer is followed by a dropout layer and a batch normalization layer to reduce the overfitting problem.

3.1.3 Activation Function

Both models use softmax as activation function instead of sigmoid because softmax is better for multiclass classification as when softmax is used, it increases the probability of one class and decreases the total probability of all other classes. This feature fits our data.

3.1.4 Optimizer

We also choose Adam as our optimizer over SGD because Adam implicitly performs coordinate-wise gradient clipping and can hence, unlike SGD, tackle heavy-tailed noise [4].

3.2 Model Maintenance and Parameter Update

Through our model tuning, we focus on the below parts: adding dense layers with dropout and batch normalization, tuning dropout rate, adding weight regularization on dense layers, and optimizing optimizers through learning rate and decay.

The dropout rate is tested from 0.25 to 1 in step size of 0.25 for Inception and 0.1 to 0.5 in step size of 0.1 for Efficient Net. The smaller dropout rate is, the better it can help decrease the influence of overfitting.

The weight regularization is added as l1 and l2 regularization to both kernel and bias. L2 regularization works better than l1 regularization.

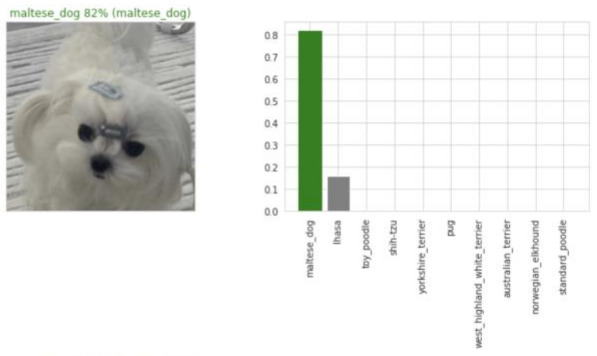
Learning rate is tested as 0.0001, 0.0005 and 0.001. Smaller learning rate helps with overfitting, but larger learning rate helps improve computation complexity. Therefore, we switch back to 0.001 when we add image preprocessing.

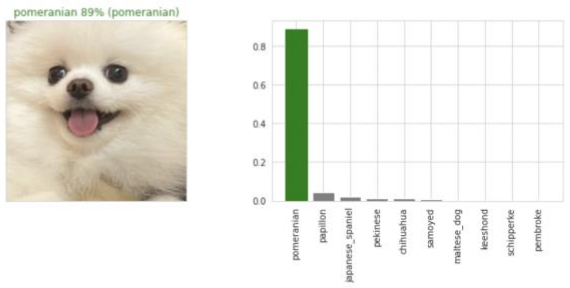
Decay is tested as 1e-6 and 1e-5. Both have improvements on overfitting, but difference is subtle.

3.3 Model Results

Four sample prediction from the efficient Net is appended below. As we can see that our model still has a rather strict requirements in the image's quality because it is hard to distinguish features like texture or relative size of the dog.

However, even the final prediction is not accurate, but we still have the right breed or similar breed across the confidence intervals.





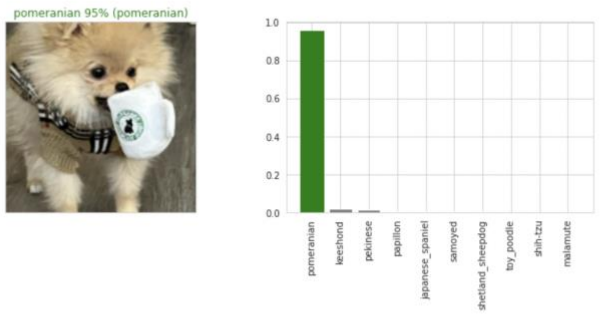


Figure 12, 13 & 14: Sample Good Prediction

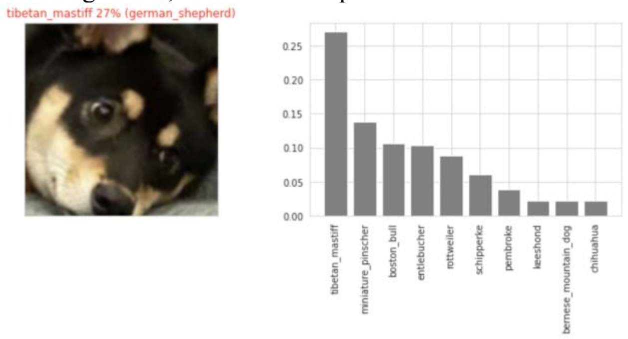


Figure 15: Sample Bad Prediction

4 Conclusion

4.1 Discussion of Findings

In this project, we want to take use of CNN's image recognition ability to identify dog breeds for us. We first fit and test our dataset with a baseline CNN but the result on validation dataset is not improving against each epoch. Also, the validation loss keeps going higher. Therefore, we believe the complexity of baseline CNN is not high enough to handle this dataset. We then try to train this dataset with Inception V3 and EfficientNet B0. Both these two neural networks have a deeper and wider design to handle more features from dog images. The initial result without any image normalization shows signals of overfitting as Inception has train accuracy 96.91% and validation accuracy 42.42%, and EfficientNet has train accuracy 99.94% and validation accuracy 64.25%. After image preprocessing with normalization, the validation accuracy increases to 78.96% for Inception and 79.49% for Efficient Net. Therefore, we find that dog breeds identification is a much more complex problem against classification between different type of animals like dog vs cat. Both Inception and EfficientNet achieve a good result but we may investigate further in the future for more improvements.

4.2 Acknowledgments

We would like to thank Professor Ashis Pujari for his teaching and answering our problems.

4.3 References and Resources

[1] Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015.

[2] Raj, B., 2022. A Simple Guide to the Versions of the Inception Network. [online] Available at: <https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202> [Accessed 30 May 2022].

[3] Kumar, A., 2022. [online] Available at: <https://vitalflux.com/keras-categorical-cross-entropy-loss-function/> [Accessed 30 May 2022].

[4] Bushaev, Vitaly. "Adam-Latest Trends in Deep Learning Optimization." Medium, Towards Data Science, 24 Oct. 2018, https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c.

[5] EfficientNet paper — Tan, M., & Le, Q. v. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. 36th International Conference on Machine Learning, ICML 2019, 2019-June.

[6] "EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling." Google AI Blog, 29 May 2019, https://ai.googleblog.com/2019/05/efficientnet-improving- accuracy-and.html.

[7] "Dog Breed Identification." Kaggle.com, 29 Sept. 2017, www.kaggle.com/competitions/dog-breed-identification/overview. Accessed 30 May 2022.

[8] SAXENA, ATRI. "Using Tensorflow 2.x to Classify Breeds of Dogs." Kaggle.com, 30 Dec. 2021, www.kaggle.com/code/atrisaxena/using-tensorflow-2-x-to-classify-breeds-of-dogs. Accessed 30 May 2022.

[9] VARGHESE, JITHIN. "Accuracy: 89.71% InceptionV3 Keras." Kaggle.com, 10 May 2021, www.kaggle.com/code/jithinanievarghese/accuracy-89-71-inceptionv3-keras. Accessed 30 May 2022.

[10] "Dog Breed CNN." Kaggle.com, www.kaggle.com/code/anu332/dog-breed-cnn.