Deep Learning for Prediction of Business Outcomes
Final Project

Improving Animal Shelter Efficiency with Neural-Networks-Based Dog Breed Detection Services

Group 25

Yiren Tang (MSBA)
Dora Zhu (MSBA)

Washington University in St. Louis
Olin Business School

Abstract

Dog breed identification is a significant challenge for animal shelters. With millions of dogs entering shelters each year in the United States, staff can only correctly identify less than 70% of them. An accurate and efficient dog breed detection system is crucial, as breed information can help fulfill a dog's specific health needs and expedite the process of finding suitable adopters for shelter dogs. Our project aims to provide a service for animal shelters that detects dog breeds using only images with deep learning techniques. Additionally, we do business with adopters by providing detailed information about their adopted dog's breed for better care. To accomplish this, we proposed using several pre-trained CNN models and extracting features from four top-performing models to build our own fusion CNN model. The results of our work demonstrate that the fusion of multiple CNN architectures achieved the best identification accuracy, with a test accuracy of 93.26%. Our approach has the potential to significantly improve the efficiency and accuracy of dog breed identification in animal shelters and will help more dogs find loving homes.

Keywords: Dog-breeds Identification; Deep Learning; CNN; Transfer Learning; Multiple-CNNs-Fusion Model

Table of Contents

1. Introduction

1.1. Background

This project aimed to detect dog breeds from images with 120 unique breeds. We applied deep learning approaches using pre-trained convolutional neural networks (CNN) and then built a fusion of multiple CNN architectures. The result shows that our Multiple-CNNs-Fusion Model performs better in identifying dog breeds with a test accuracy rate of 93.26%.

1.2. Motivation

Our work is highly applicable to real-world animal shelters and is therefore worth investment. According to the ASPCA, about 3.3 million dogs enter shelters every year. [1] However, a study from Plus One shows that shelter staffs correctly identify a dog's breed only 67% of the time. [2] Identifying the breed correctly while efficiently is important for the shelters, the shelter dogs, and also those adopters. With our detection service, shelters can quickly fulfill dogs' health needs as some illnesses are breed-specific. Additionally, streamlined breed detection can help shelters save costs and allocate resources more effectively, ultimately aiding more stray dogs in need. Moreover, our solution expedites the process for dogs to find suitable adopters. From ASPCS, 1.6 million dogs are adopted from shelters every year, which accounts for 23% of all dogs. [3] Accurate breed information listed on the shelter's website allows adopters to make informed decisions, as knowing the breed means understanding the dog's behavior better and reducing the chance of second-time abandonment.

The investment earns revenue by fee-charging from animal shelters and individual adapters. We will charge animal shelters an annual fee for subscribing to our breed identification service. Adopters will have the option to pay a one-time fee for receiving more detailed information about their dog's breed, including behavior tendencies, preferences, and health instructions. This information will be provided to them in a comprehensive manual that will assist them in taking proper care of their adopted dog.

2. Review of the literature

2.1 Related work

Dog breed identification is a common topic in the machine learning area. One of the earlier works in dog breed classification was a 2012 paper "Dog Breed Classification Using Part Localization" by Liu et. al. [4] The key takeaway in this paper was that fine-grained classification can be improved if the features used for classification are localized at object parts. Liu et. al built an SVM regressor with greyscale SIFT descriptors as features to isolate the face of the dog. With this part localization technique, Liu et. al achieved a 67% recognition rate on a dataset containing 8,351 images with 133 dog breeds.

In 2018, Ráduly et. al in "Dog Breed Identification Using Deep Learning" proposed convolutional neural networks used in dog breed determination. [5] They introduced transfer learning, and results showed that NASNet-A mobile architecture achieved 80.72% test accuracy while the Inception-ResNet-v2 network

achieved 90.69% on the test dataset. This study proves the raise of accuracy using CNN on the dog breed classification topic.

In 2018, Uno et. al launched their paper “Comprehensive Study of Multiple CNNs Fusion for Fine-Grained Dog Breed Categorization”. [6] They proposed the fusion of multiple CNN architectures, specifically, the fusion of different layers in AlexNet and VGG-16. This study proves the improvement of the fusion model over a single VGG-16 on test accuracy, from 81.2% to 84.08%.

2.2 Limitation of the existing system

Applying the CNN model to identify dog breeds is common and can achieve high enough test accuracy. While it is insightful that Uno et.al brought up a new idea that could further increase the model performance. However, their fusion model only combined two pre-trained models, AlexNet and VGG-16. There was no evidence showing AlexNet and VGG-16 performed best among all the transfer learning, and accuracy may increase if we train the fusion model with more pre-trained models. Therefore, our work would be unique and meaningful in exploring deeper on the topic of the fusion of CNN architectures.

3. Problem description

Our work focuses on applying a fusion CNN model to detect dog breeds given dog face images. One main challenge is deciding which model and the number of models we should use in the fusion model.

For the model choices, we first trained our own convolutional neural network but turned out the accuracy was extremely low. Considering GPU and RAM limitations and the time consumed, we decided not to train a more complex model completely ourselves. Instead, we apply several pre-trained models and build a fusion model using features extracted from them. In the experiment, we compared six pre-trained models and ranked the performance by test accuracy results. For the number of models used, we found using four models was proved to have the best test accuracy for the fusion model during our experiment. Therefore, for the final fusion CNN model, we chose four pre-trained models that stood up in performance and combined the features extracted by these four models for training the fusion model.

Our work builds upon previous research that used convolutional neural networks to identify dog breeds. Specifically, we focused on training a fusion CNN model to further improve accuracy. Compared to previous researchers, we placed greater emphasis on selecting those top-performance models and determining the optimal number of models to use in our fusion model. As a result, our work demonstrated improved experimental performance in accurately identifying dog breeds. Our approach contributes to the growing body of research on the use of deep learning in dog breed identification and will significantly benefit animal shelters and adopters.

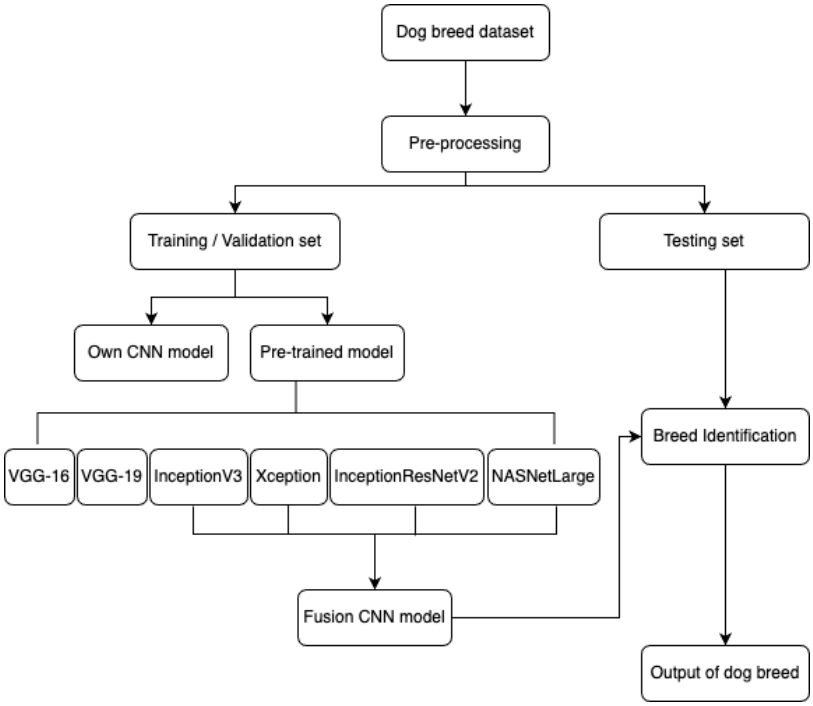

Figure 1: Problem Framework

4. Model description

The purpose of this subtopic is to provide a concise overview of the methods used during the experiment. The stepwise implementation of the models and further improvement would be elaborated in section 5.4, the 'Experiment results and parameter optimization'.

4.1 Basic Convolution neural network (CNN)

Dog breed identification is a case of image processing, and convolutional neural networks (CNNs) are commonly used in analyzing visual images. For this reason, we initially developed our own CNN model to evaluate its efficacy. CNNs comprise convolutional layers, pooling layers, and fully connected networks. The convolutional layer extracts features, the pooling layer reduces the output matrix size, and the fully connected network classifies the input. Using the CNN technique, we can process dog images and generate classification results of dog breeds.

4.2 Pre-trained CNN model

In our experiment, to solve the problem of low test accuracy generated by a self-built CNN model while avoiding GPU and RAM limitations, we chose to implement pre-trained CNN models. Pre-trained models are usually created and trained by tech giants like companies such as Google, Apple, and Amazon. We are able to use their models since we are solving similar problems, and they use a very large dataset so the models would be robust.

VGG 16 and VGG 19
These two models were developed by the Visual Geometry Group at Oxford University. They are named given the number of weight layers in the network. That is, VGG16 has 16 layers and VGG19 has 19 layers.

InceptionV3 and Xception
These two models were developed by Google. InceptionV3 uses multiple filter sizes in parallel, while Xception extends on the basis of Inception architecture with depthwise separable convolutions. Generally speaking, Xception allows for more efficient computation and fewer parameters.

InceptionResNetV2
This is a hybrid architecture that builds on the Inception family of architectures with residual connections.

NASNetLarge
This is a neural architecture search architecture created by Google. It uses a reinforcement learning algorithm to search for the optimal parameters of the given search space of filters, number of layers, etc. NASNetLarge is one of the most powerful pre-trained models, but with a drawback that requires large computation power.

5. Data and experimental results

5.1 Data collection

We obtained our dataset from Kaggle (https://www.kaggle.com/competitions/dog-breed-identification), which consists of 20,579 images of 120 different dog breeds. The dataset is divided into a training set, which contains 10,222 labeled images, and a test set, which contains 10,357 unlabeled images. As a result, we will only use the labeled images in the training set to train our model.

Among the 120 breeds in the dataset, the breed with the most images is the Scottish deerhound, which has 126 images. The breed with the fewest images is the Eskimo dog, which has 66 images. However, even the smallest breed category has over 60 images, so we believe the dataset is sufficiently large for training our model.

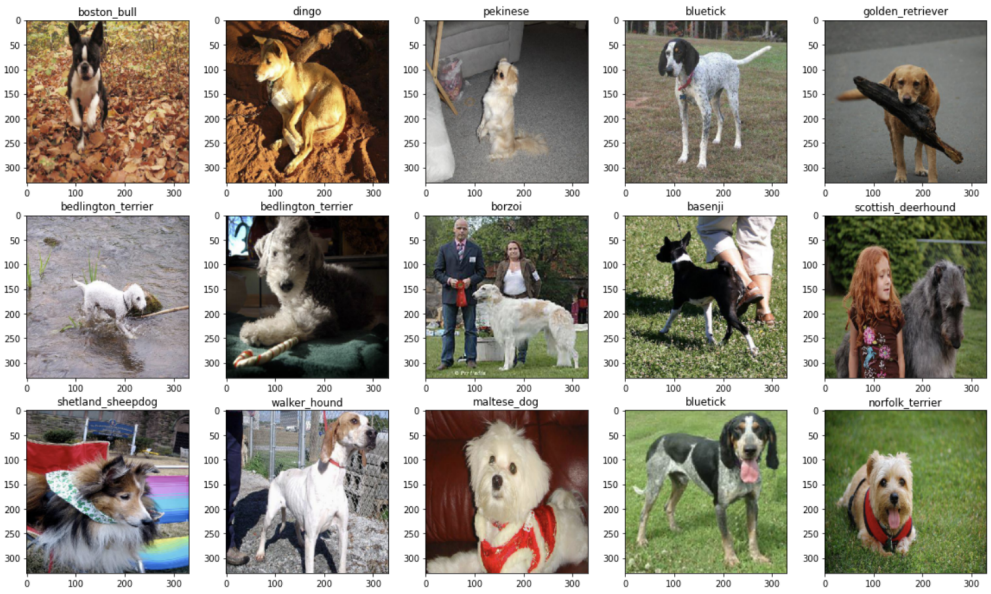

Figure 2: Sample of Images

5.2 Data processing

To begin, we set the image shape to 331x331x3. Next, we assign a label to each image in our dataset. Since the output of our predictor for each input is a vector of probabilities for each class, we will use one-hot encoding to represent the labels. This means that for each input, we will create a row vector of length 120 with a 1 at the index corresponding to the label and 0's everywhere else. To simplify this process, we define a function that will perform the one-hot encoding for us.

5.3 Train/Validation split

We will reserve 10% of our images for testing purposes. The remaining images will be split into a training set and a validation set using an 80/20 split. This means that 20% of the remaining images will be used for validation during the model training process. After splitting the data, our training set, validation set, and the test set will contain 7,459, 1,840, and 1,023 images, respectively.

5.4 Experiment results and parameter optimization

We first trained our own convolutional neural network. The structure of the model is shown in Figure 3. We set the shape of input images at 224*224*3. The first layer is a Conv2D layer with 64 filters, a kernel size of 5x5, and a ReLU activation function. The next layer is a MaxPool2D layer with a pool size of 2x2, which reduces the size of the feature maps by taking the maximum value within each 2x2 region. A Dropout layer is added to prevent overfitting by randomly setting 25% of the input to 0 during training. The same pattern of Conv2D, MaxPool2D, and Dropout layers is repeated three more times with different filter sizes (32, 16, and 8) and kernel sizes (3x3, 7x7, and 5x5) to extract increasingly complex features from the input images. After the convolutional layers, a Flatten layer is used to convert the 3D feature

maps into a 1D vector. Two Dense (fully connected) layers with 128 and 64 units and ReLU activation functions are added for classification. Dropout layers are added after each Dense layer to prevent overfitting. The final layer is a Dense layer with a softmax activation function, which outputs a probability distribution over the possible classes. The model is compiled using categorical_crossentropy loss, Adam optimizer with a learning rate of 0.0001, and accuracy as the evaluation metric. The kernel_regularizer parameter is used in the Conv2D and Dense layers to apply L2 regularization to the kernel weights, which helps prevent overfitting.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 220, 220, 64)	4864
max_pooling2d (MaxPooling2D)	(None, 110, 110, 64)	0
dropout (Dropout)	(None, 110, 110, 64)	0
conv2d_1 (Conv2D)	(None, 108, 108, 32)	18464
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 32)	0
dropout_1 (Dropout)	(None, 54, 54, 32)	0
conv2d_2 (Conv2D)	(None, 48, 48, 16)	25104
max_pooling2d_2 (MaxPooling2D)	(None, 24, 24, 16)	0
dropout_2 (Dropout)	(None, 24, 24, 16)	0
conv2d_3 (Conv2D)	(None, 20, 20, 8)	3208
max_pooling2d_3 (MaxPooling2D)	(None, 10, 10, 8)	0
dropout_3 (Dropout)	(None, 10, 10, 8)	0
flatten (Flatten)	(None, 800)	0
dense (Dense)	(None, 128)	102528
dropout_4 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8256
dropout_5 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 120)	7800

Figure 3: Structure of Self-Designed CNN Model

The performance of the model is displayed in Figure 4. While there is no evidence of overfitting, both the train accuracy and validation accuracy are extremely low. Moreover, the test accuracy is merely 3.13%. These results suggest that a more complex model may be necessary to achieve a higher accuracy rate. However, due to limitations in GPU and RAM, it is difficult for us to train a more complex model with over 9000 pictures. As an alternative, we plan to use several pre-trained models to extract features from these images, and then use these features to train our own model.

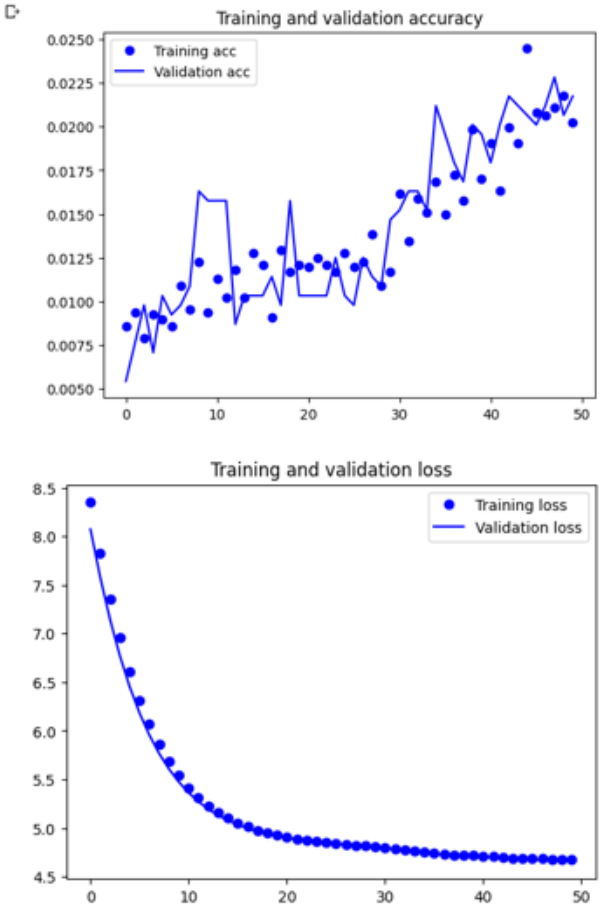

Figure 4: Performance of Self-Designed CNN Model

To achieve better results with pre-trained models, we started by running six pre-trained models to independently extract features and train our model based on these features. The performance is summarized in Table 1. As shown in the table, VGG16 and VGG19 achieved an accuracy of around 74%, while the remaining four models (InceptionV3, Xception, InceptionResNet, and NASNetLarge) achieved an accuracy of around 90%. Among these four models, NASNetLarge performed the best with an accuracy of 92.67%. Therefore, we combined the features extracted by InceptionV3, Xception, InceptionResNet, and NASNetLarge to train a fusion model with the aim of achieving higher accuracy.

Model	Accuracy	Feature Map Shape
VGG16	73.41%	(9199,512)
VGG19	74.10%	(9199,512)

InceptionV3	88.17%	(9199,2048)
Xception	89.35%	(9199,2048)
InceptionResNetV2	91.50%	(9199,1536)
NASNetLarge	92.67%	(9199,4032)
Fusion Model	93.26%	(9199,9664)

Table 1: Accuracy of Different Models

The fusion model will include 9644 input features. To handle the large 9664 input features, we designed our model with three dense layers. The first layer has 1,028 nodes and uses the ReLU activation function. The second layer has 128 nodes and also uses ReLU activation. The final layer has n_classes nodes, corresponding to the number of unique dog breeds in our dataset, and uses the softmax activation function to output a probability distribution over the 120 classes.

To improve model performance, we implemented a learning rate annealer that adjusts the learning rate based on validation accuracy. If validation accuracy doesn't improve for three epochs, the learning rate decreases by 0.01. The minimum learning rate is 1e-5. Additionally, we added early stopping to increase operational efficiency. If validation loss doesn't improve for 10 epochs, training stops.

The outcome of the first model is displayed in Figure 5. The test accuracy rate is 92.38%. We observed that training accuracy significantly exceeded validation accuracy, indicating there existed an overfitting problem.

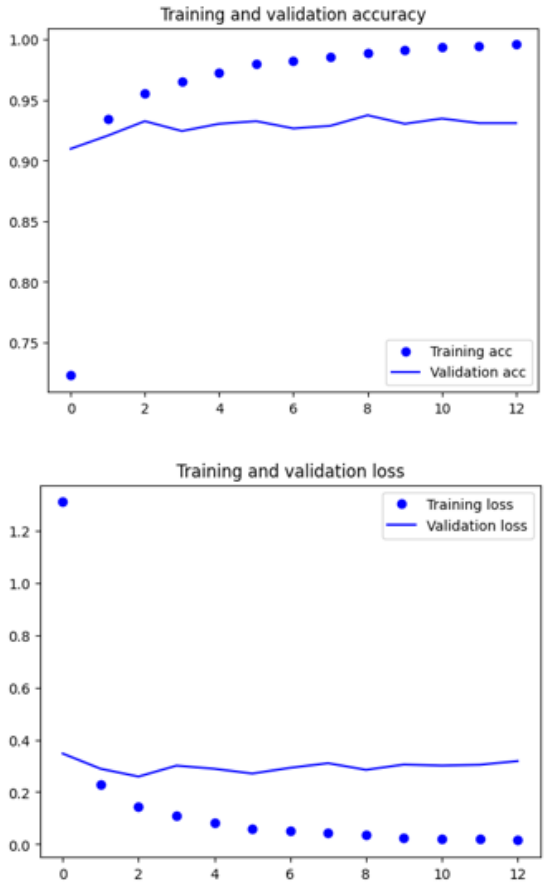

Figure 5: Training and Validation Comparison of The First Model

To address the overfitting problem, we added two dropout layers between the dense layers with a rate of 0.5. Dropout is a regularization technique that randomly drops out a percentage of the nodes in the layer during each training iteration, which helps prevent overfitting by reducing interdependence among the neurons. The first dropout layer is placed after the first dense layer, and the second one is placed after the second dense layer. In addition to dropout, we also apply L2 regularization to the first dense layer to add an additional penalty term to the loss function, which helps prevent overfitting by discouraging large weights in the model. Finally, we re-apply the learning rate annealer and early stopping to the model. With these adjustments, we hope to improve the generalization performance of the model and prevent overfitting.

The performance of the second model is displayed in Figure 6. The test accuracy of the model decreased to 91.1%. While the overfitting issue has been addressed, the validation accuracy surpasses the training accuracy by a significant margin, indicating the problem of underfitting.

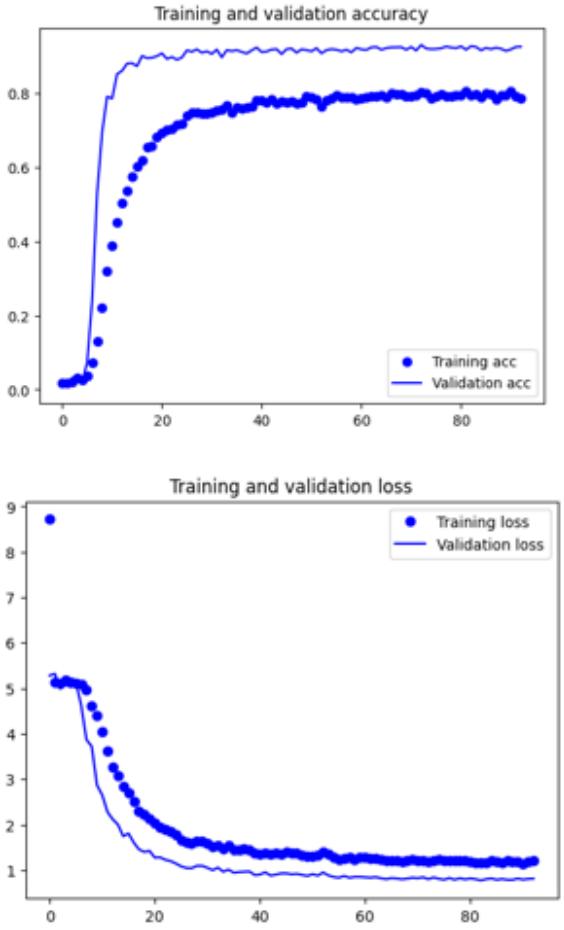

Figure 6: Training and Validation Comparison of The Second Model

To address the underfitting problem in the previous model, we made some modifications to the architecture. Specifically, we removed L2 regularization from the first layer and increased the number of nodes in the second layer to 256. We also kept the two dropout layers with a rate of 0.5 to prevent overfitting. The resulting model is the third model and its performance is shown in Figure 7. This model achieved a test accuracy of 93.26% and both the training and validation accuracy are almost the same, indicating that there are no overfitting or underfitting problems in this model.

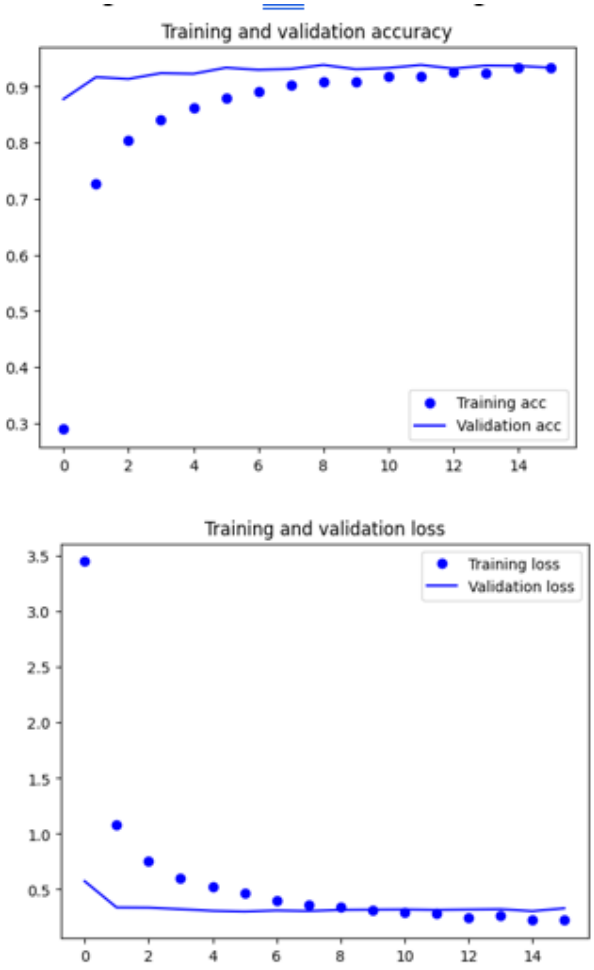

Figure 7: Training and Validation Comparison of The Third Model

6. Conclusions, discussions, and recommendations

Our fusion CNN model combined features extracted from InceptionV3, Xception, InceptionResNet, and NASNetLarge. By leveraging the learning rate, early stop, and dropout layers, this model achieved a test accuracy of 93.26%. Compared with the basic CNN model built ourselves, such a result was a significant improvement. In addition, the technique of the fusion architecture manifested 0.59% raise in test accuracy over the best performance of using only one CNN pre-trained model: NASNetLarge. Although 0.59% is not a large change in the scale of number, we still considered this improvement a meaningful stage to the real-world shelter dog breed detection cases as our model increased the accuracy even closer to no error.

For 3.3 million dogs entering animal shelters in the U.S. every year, we are able to provide a service that correctly identifies breeds for more than 3 million of them in less than one minute. While this service generates revenue for investors, its impact on saving dogs' lives is immeasurable. Our service, based on the fusion CNN model, will help animal shelters raise the accuracy of breed identification from 67% to an

impressive 93.26%. This means that more dogs can receive breed-specific health checks without long waiting times, improving their chances of survival. Moreover, with our service, these dogs have a higher likelihood of finding suitable homes, creating a win-win solution for both the dogs and their adopters.

To further improve our service, we plan to train our model to take into account mixed-breed dogs. The updated version of our service aims to identify not only the predominant breed but also the second and third most prominent breeds. Our goal is to provide shelters and adopters with even more comprehensive information about the dogs in their care. With this added level of accuracy, we believe these dogs can receive the best possible care and find loving homes where they can thrive.

Reference

[1] “Pet Statistics.” *ASPCA*, www.aspca.org/helping-people-pets/shelter-intake-and-surrender/pet-statistics.

[2] Gunter, Lisa M., et al. “A Canine Identity Crisis: Genetic Breed Heritage Testing of Shelter Dogs.” *PLOS ONE*, vol. 13, no. 8, 2018, https://doi.org/10.1371/journal.pone.0202633.

[3] “Pet Statistics.” *ASPCA*, www.aspca.org/helping-people-pets/shelter-intake-and-surrender/pet-statistics.

[4] J. L. et al. Dog breed classification using part localization. Computer Vision: ECCV 2012, pages 172–185, 2012.

[5] Z. Ráduly, C. Sulyok, Z. Vadászi and A. Zölde, "Dog Breed Identification Using Deep Learning," *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*, Subotica, Serbia, 2018, pp. 000271-000276, doi: 10.1109/SISY.2018.8524715.

[6] M. Uno, X. -H. Han and Y. -W. Chen, "Comprehensive Study of Multiple CNNs Fusion for Fine-Grained Dog Breed Categorization," *2018 IEEE International Symposium on Multimedia (ISM)*, Taichung, Taiwan, 2018, pp. 198-203, doi: 10.1109/ISM.2018.000-7.