A machine learning study of Dog Breed

Identification ANLY 535 - Group 71

Rahul Awale, Rajendra Naidu Mannam, Sandeep Joshi, Jiawen Wei, Li Wang, Zengxin Qin

1 Harrisburg University of Science and Technology

2 ANLY 535 – 50 – A – 2020/Summer - Machine Learning II

Professor Roozbeh Sadeghian Phd

August 7, 2020

Author Note

This is a research study for ANLY 535 class.

Correspondence concerning this article should be addressed to ANLY 535 - Group 7, 326 Market St, Harrisburg, PA 17101.

Abstract

It is a very interesting vision identification problem for machine learning or deep learning methods to identify the dog breed based on some data and images. We got this data and images from Kaggle and we will randomly pick 2 breed classes to do some design on binary classification. For example, we will choose Siberian Husky and Golden Retrievers to plan some tasks on binary classification. There will be 120 classes with all kinds of challenges such as how to encode the different breed labels and so on. For example, we will use what we learned from the lecture such as vectors: [0,0, . . . ,0,0,1, . . . ,0]. We will design models and try to improve their accuracy during this project.

*Keywords:* Dog Breed Identification, Classification, Convolutional Neural Network

Word count: 2,139

# 1Introduction

The main goal of our project is to identify different dog breeds from images. Many times, people have mistaken one dog breed for another. When it comes to health issues, this could be a significant problem as many illnesses are common on certain breeds as opposed to others. We believe analytics could help solve this problem. So, to make things better, we plan to use many dog images and apply machine learning techniques (such as CNN) to create an algorithm that can help identify the correct dog breed.

When it comes to this project, we do have some challenges. First of all, this is a fine-grain classification problem. What that means is that all dogs hare very similar bodies and facial features. For example, a Morkie and Maltipoo can look identical, making it difficult to differentiate between the two. Second, dogs have fewer differences between breed and more differences within breeds such as size, color, shape, etc. Third, dogs are the largest diverse species on earth, and it will be challenging to identify every breed that exists. Lastly, the photographs we use have the same dog breed in different positions, lightings, and background. We also have other issues, such as having two same breed dogs in one picture. All these problems with the photos can increase the difficulty in identifying the correct dog breed.

We chose this problem because it solves other fine-grained classification problems too. For example, instead of dogs, the same algorithm can be used to identify cars, birds, phones. Next, it can help researchers and scientists save time and resources. For example, people involved in any identification study can process their work much faster than traditional methods. Similarly, regular healthcare workers and veterinarians can also benefit from this. For example, suppose someone gets a snake bite, and they have a picture of the snake. In that case, the doctor can instantly use this algorithm to identify if the snake was venomous or not and treat the person accordingly. The possibility is endless. Lastly, we chose dogs because they are the most popular animal on the internet. Therefore, we can find thousands of dog images easily.

# 2 Related Work

The first is a mobile application called Dog Scanner. It allows users of the app to use the camera on their phone to upload a picture. It goes through a set of algorithms in the backend and gives them the result fast and easy. It even identifies mixed breed and tells the user what kind of mix breed their dog is (Dog Scanner -#1Dog Breed Identification on Android and iOS,2020). To make data collection more accessible and accurate, they have also added some gamification to their app. The goal of the game is to catch every type of dog. It allows the company to gather more dog data and further improve their algorithm. Also, similar apps are emerging. Another application is called Fetch, which uses artificial intelligence to analyze and identify a dog by its breed using a smartphone camera or a photo library (Shea, 2018). All these applications are similar to the goal we are trying to achieve in this project.

# 3Data and Methodology

### 3.1.1 Data Source Description

We sourced our data from kaggle (dog-breed-identification, 2017). It consisted of 10222 unique images of 120 different breeds of dog. Each image bears a unique hashed ID which is used in a label file to annotate the breed.

There is median of 82 sample images per dog breed.

### 3.1.2 Pre-Processing

Out of 120 breeds in the dataset (dog-breed-identification, 2017), we restricted our study to two breeds – Shetland Sheepdog and Golden Retriever.

|  |  |
| --- | --- |
|  |  |



Figure 1.a. A Shetland Sheep dog v/s b. a Golden Retriever

This choice was made as these breeds presented a quite balanced classification problem as they seem similar to an untrained eye but they differ enough that our relatively smaller training dataset would suffice.

A preliminary classification endeavor made using pre-trained CNN model ResNet50(MathWorks, n.d.) gave us an accuracy of 62.93% out of the box. This model has over 50 layers as is trained on over million images!

The secondary rational behind narrowing our classification problem to two breeds was that it enables us to manually assess the pictures and choices our model made.

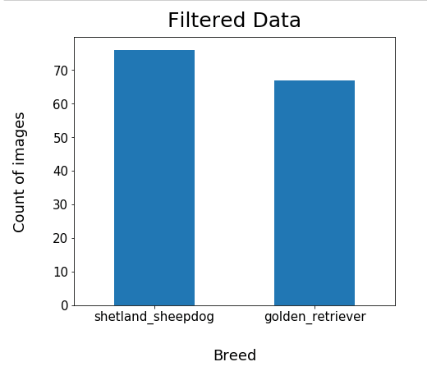


Figure 2 Filtered Dataset

We used cv2(opencv-python, n.d.) package to read, compress and resize our images. Such down sampled renders are less processing intensive while retaining all the important features.

Most of the image processing studies employ 128 by 128 pixel size and we also decided to go with this sizing as it provides good accuracy with respect to memory and computing units consumed.

|  |  |
| --- | --- |
|  |  |

Figure 3.a. Original sample image on left; 3.b. downsampled image using cv2 package

We also used same package to train model on gray scaled samples as from the study – ‘Dog Breed Identification’ (W. LaRow, 2016) found that certain breeds were misclassified due to different colored dogs and CNN models may suffer from associating pixel color with more weight.

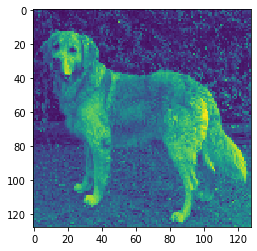


Figure 4. A downsampled Gray-scale version of sample in figure 3. Image has been resized to 128 x 128 p.

## 3.2 Methodology

## 3.2.1 - Model Design

In this study, we have designed Convolutional Neural Network (CNN) to train the dog-bread dataset and identify the dog bread between Shetland Sheepdog and Golden Retriever. CNN contains forward propagation, loss, and back propagation stages where Forward propagation contains three layers which are convolution, pooling and fully connected layers. We have used trial and error method to select the multiple convolutional layers and pooling layers during the design process and finalize the one model which predict dog breed more accurately.

Final CNN model contains below list of layers

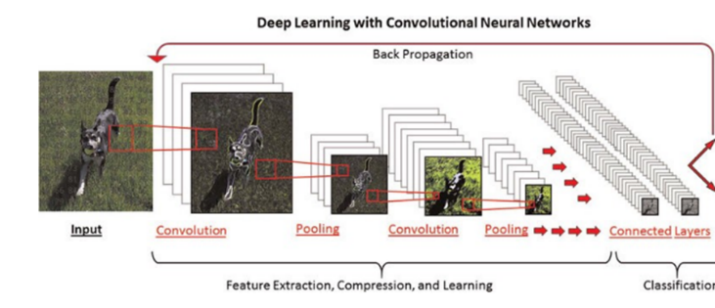
One input layer

Two convolutional layers

Two Max Pooling layers

One Flatten layer

One Dense layer



HOW CNN Algorithm Works: when we trained the model with Dog bread dataset, it follows below 5 steps.

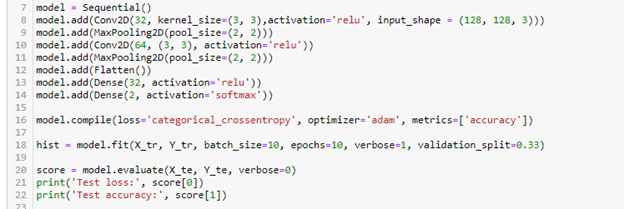
Step 1: Initialize all filters and parameters / weights with random values

Step 2: Import a training image as input, which goes through the forward propagation step (convolution, ReLU and pooling operations along with forward propagation in the Fully Connected layer) and find the output probabilities for each class

Step 3: Compute error (loss)

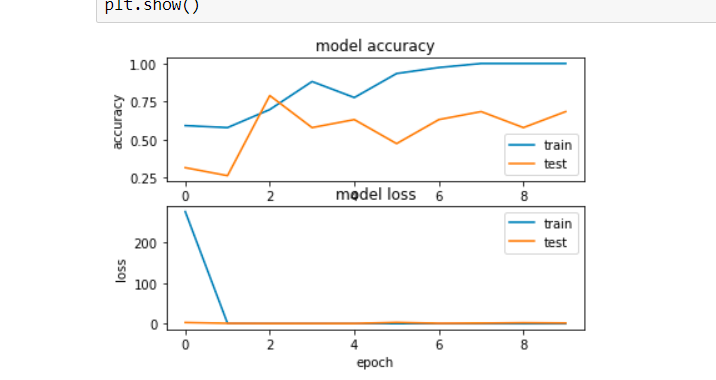
Step 4: Use backpropagation using gradient descent to update all the parameters.

Step 5: Repeat steps 2~4 for all the training instances



Dog data set contains images (128\*128) which is used as input to the CNN model, in the first Convolutional layer, we have selected the 32 filters with kernel size as 2\*2 matrix which should be passed through activation function called ‘relu’ for non-linearity. Model learns the values of 32 filters and extract 128\*128\*32 features once image passed through first convolutional layer. The output of Conv layer passed to the Max pooling layer where it down sample the values to 64\*64\*32 which contains all image information. Repeated the conv layer and Max pooling layer one more and pass it to next layer that is flatten layer where it converts output of max pooling layer to one long vector which is used by dense layer. Here dense layer act as classifier. Next, compiled the model with categorical cross entropy and identified the loss. Applied back propagation using Adma optimizer to update weights. We trained model with training data with batch size as 10 and epochs as 10 then run the predictions.

3.2.2- Model Prediction

The first epoch took some time to run but the rest ran considerably faster. The model was executed on 76 training samples with 10 epochs. The overall test loss is around 2.897 and test accuracy is around 0.5 to 0.6. On the training accuracy, the model started with 0.59 and 0.58 on the 1st and 2nd epochs, and gradually increased to 0.6974 on the 3rd epoch. Starting from 3rd epoch, the accuracy spiked to 0.77 on epoch 6 and jumped significantly to 0.93 on epoch 7. And gradually reached the top with 1. The training loss does not look good on the 1st epoch where it was 275, then dropped significantly to around 1.0 starting with epoch 2 and gradually decreased to close to 0 as epoch runs. Overall, the result looks ok and we are going to try out the model with dropout to see if the performance could be enhanced dramatically. 

### 3.2.2 Convolutional Neural Network with Dropout

A CNN model with dropouts were implemented to avoid overfitting problem. This technique uses only partial outputs from previous pooling layer and fully connected layer, thus avoiding potential over training. The model adopted dropouts at both before and after fully connected layer at 0.6 rate.

After applying 0.6 dropouts before and after fully connected layer, model training and testing receive similar accuracy at around 4th epoch. Model accuracy is 0.67.

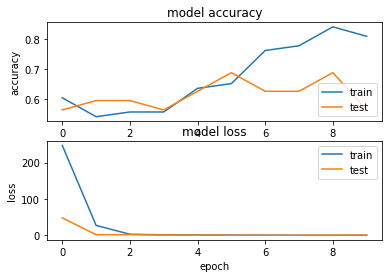


Figure 1 Model accuracy and loss

# Conclusion

Our goal of this project is to create a model/classifier to identify a dog’s breed from an image. We have used a deep learning and CNN (Convolutional Neural Networks) in this dog breed detection. We consider that our results to be success after we evaluated our model. We are able to use our neural network to predict dog’s breed with the dataset. We have tested the accuracy of our model and classification algorithms for detecting dog’s breed.

# Limitation

There are also some limitations we have noticed during the project. For example, we had some difficulties when there are more than one dog in the image. It might cause some bad prediction or results. And secondly, when we were trying to improve our model by adding more layers, the processing time will significantly to up. Eventually, it can take hours to run or process when we add more changes.

# Future Work

In the future, based on our success of this detection work, we would like to believe this is a promising method for future studies. The neural network takes too much time to train and we were unable to provide more iterations on our project. Our recommendation for future research will be train these models or neural networks with a different batch iterator and architecture to see if we have a better result.

# References

1. *Dog-breed-identification*. (2017). Retrieved from Kaggle.com: <https://www.kaggle.com/c/dog-breed-identification/data>
2. MathWorks. (n.d.). *ResNet50*. Retrieved from MathWorks: <https://www.mathworks.com/help/deeplearning/ref/resnet50.html>
3. *Opencv-python*. (n.d.). Retrieved from <https://github.com/skvark/opencv-python>
4. W. LaRow, B. M. (2016). *Dog breed identification*. Retrieved from Network: <https://web.stanford.edu/class/cs231a/prev_projects_2016/output%20(1).pdf>
5. Dog Scanner - #1 Dog Breed Identification on Android and iOS. (2020, May 22). Retrieved August 06, 2020, from <https://siwalusoftware.com/dog-scanner/>
6. Shea, E. (2018, January 22). New App Uses Artificial Intelligence To Identify Dogs By Breed. Retrieved August 06, 2020, from <https://www.akc.org/expert-advice/news/app-identifies-dogs-by-breed/>

# Annexure

1. Classification using ResNet50:

from keras.applications.resnet50 import ResNet50

from mpl\_toolkits.axes\_grid1 import ImageGrid

from keras.applications.vgg16 import preprocess\_input, decode\_predictions

from keras.preprocessing import image

from keras.preprocessing import image

def read\_img(img\_id, size=(256, 256)):

img = image.load\_img(os.path.join(image\_path, img\_id + '.jpg'), target\_size=size)

img = image.img\_to\_array(img)

return img

model\_res = ResNet50(weights='imagenet')

NUM\_CLASSES = 16

j = int(np.sqrt(NUM\_CLASSES))

i = int(np.ceil(1. \* NUM\_CLASSES / j))

fig = plt.figure(1, figsize=(16, 16))

grid = ImageGrid(fig, 111, nrows\_ncols=(i, j), axes\_pad=0.05)

for i, (img\_id, breed) in enumerate(df.values[:16]):

ax = grid[i]

img = read\_img(img\_id)

ax.imshow(img)

x = preprocess\_input(np.expand\_dims(img.copy(), axis=0))

preds = model\_res.predict(x)

\_, imagenet\_class\_name, prob = decode\_predictions(preds, top=1)[0][0]

ax.text(10, 180, 'ResNet50: %s (%.2f)' % (imagenet\_class\_name , prob), color='w', backgroundcolor='k', alpha=0.8)

ax.text(10, 200, 'LABEL: %s' % breed, color='k', backgroundcolor='w', alpha=0.8)

ax.axis('off')

plt.show()