CNN Project: Dog Breed Classifier

By

Rizwan Mushtaq

Project submitted in partial fulfillment of the requirements for the award of the nanodegree of

Machine Learning Engineer

In the month of March 2021

Submitted to:

UDACITY

1. Introduction

This project will be dealing with the development of the algorithm to predict the dog breed by taking user provided images as input. This project takes central place in learning Artificial Intelligence domain. Since it deals with a very complex task i.e., the identification of the dog breed. The classification problem such as classifying human or dog is somewhat easy as compared to classification problems that deal with identifying dog breed and human race. The latter problem requires complex algorithms, careful analysis and huge amount of training data.

In this project we will be dealing with the classification problem, more specifically image classification problem. In order to solve that problem the best instrument available is CNN convolutional neural network. The CNN is an algorithm that has gained the reputation over the years to classify images. It was originally introduced by Yann LeCun in 1980s to classify low resolution images. However, latter it became popular and is being used in wide range of industries, from hyperspectral imagery see for instance (Hu et al., 2015), medical science (Lundervold & Arvid, 2019) to satellite imagery classification (Maggiori et al., 2016). For a comprehensive application and developments of Deep Convolutional Neural Networks for Image Classification, see for example (Rawat & Wang, 2017).

The results of this project are encouraging because it identifies correctly dog breeds. These algorithms could also be used in other situations such as identifying human races, categorizing male, female or classifying humans like kids, young and older by utilizing corresponding data. Today, we use to see the application of this project in every day's life, security cameras might recognize the individuals and their background by just analyzing their faces. We also see several facial recognition apps such as Face ID in latest iPhone. All these reasons motivated me to choose this topic to dive deeper in this project and I literally enjoyed doing each step of the project.

2. Problem Statement

The problem we are facing here is, whether we can identify dog breeds by using Artificial Intelligence algorithms? Since it is a hard task even for humans to identify the correct breed of the dogs. We'll examine whether algorithms can do the task efficiently and effectively or not. In this project we attempt to predict the dog breed by providing the algorithm an image of the

dog and the algorithm returns the type of dog breed. The solution strategy to this problem is to adopt and apply a Deep Convolutional Neural Networks for Image Classification. Thes CNN is well known for its power and capacity to classify images, for more on CNN see for instance (Rawat & Wang, 2017). This is the best available option for this type of problem that we are dealing with.

3. Evaluation Metrics

Since we are dealing with a multi-class problem, it is therefore good idea to use the accuracy matric. This matric is used to measure the performance of the underlying model. The accuracy matric can be represented as:

$$Accuracy = Number\ of\ correclty\ classified\ images/Total\ Images$$

The above expression is the ratio of correctly classified images out of total number of images. The matric we adpoted here is optimal for this dataset because the classes for this dataset nearly balanced. With nearly balanced dataset classes, we may proceed with the accuracy matric.

4. Data Exploration

The objective of this analysis is to predict the dog breed. Therefore, data required for this objective is the images of dogs. The underlying dog dataset contains a total of 133 breed/categories of dogs and 8351 dog images. In addition to that the human dataset includes 13233 human images. We observe that both datasets are unbalanced, since we do not have equal categories for dog as well as for humans. For the purpose of training, validation and testing we further divided the data into three categories i.e., training (6680 images), testing (836) and validation (835).

It is also worth mentioning here that we have over 8000 images of dog, these images are not taken by professionals and not with the same frame. Therefore, there are images with bad light conditions, inappropriate backgrounds or perhaps dog posture in the photo is not clear, size of the images is also different. These all issues might affect the performance of the model. Similarly, human images dataset also has these issues. However, human files have almost same size i.e., (250 x 250).

5. Data Visualization

As described above, we have a sufficient amount of data for the proposed model. Our objective is to predict the dog breed and we have 133 different dog breeds in our dataset. While each breed contains sufficient images in training_target as shown in below figure . Where we can observe minimum number of images in a class is (around 27) and maximum number is 76.

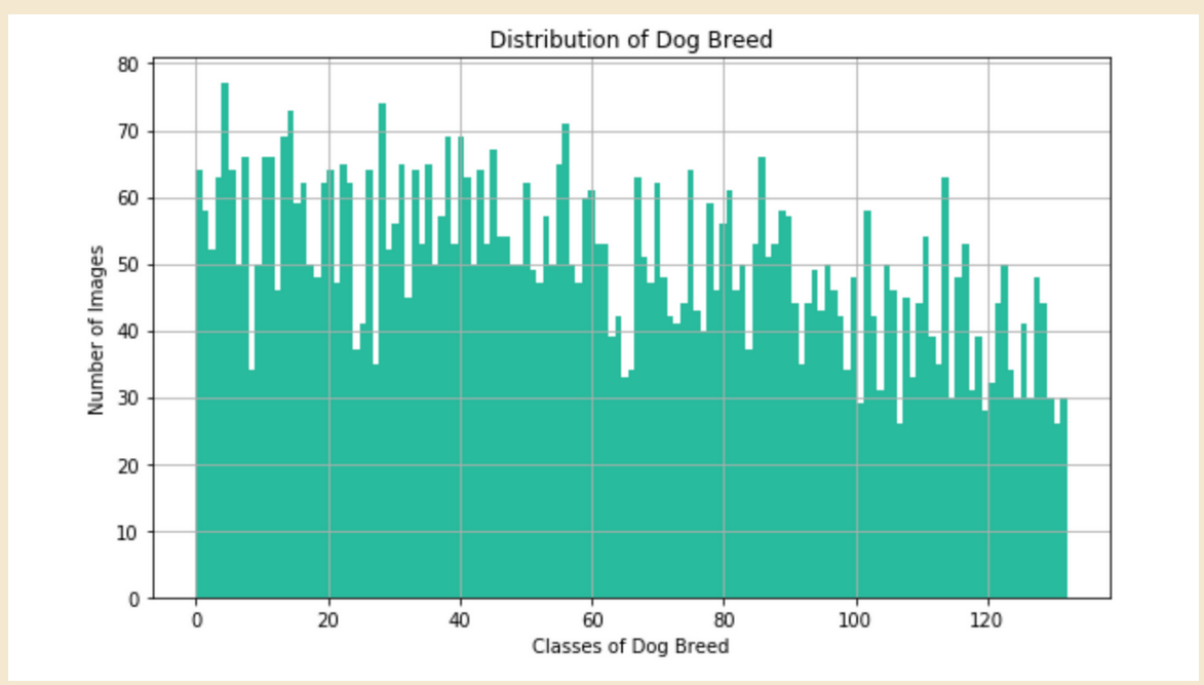


Figure: Dog Breed Data Visualization

6. Algorithms and Techniques

Main objective of this project is to build the model that identifies a dog breed and to identify the input image whether it is a human and if human were a dog he belongs to which dog breed. Considering the underlying multiclassification problem the most appropriate model to be applied here is CNN convolutional neural network. Conventional face recognition methods have been surpassed by deep learning methods based on convolutional neural networks (CNNs) with high accuracy and robustness acquired by learning from actual deviations appearing in the images. The CNN models are being widely used in image recognitions from satellite imagery to medical sciences and for security purposes (facial recognition), and the list goes on. Therefore, in this project I also utilizes the CNN convolutional neural network using Pytorch library. The process this project adopts is as follows: in the first step it identifies the human

image, secondly, it identifies the dog images and in the last step it identifies the dog/human first then their corresponding breeds as defines earlier. Possible outcomes of this algorithm are:

- If a dog is detected in the image, it returns its corresponding breed
- If a human is detected in the image, it returns most resembling dog breed.
- In third situation if neither is detected it may return an error message.

7. Benchmark Model

For the sake of benchmarking, we will restrict the CNN till 20 Epochs and extend by using Transfer learning which trains and validates the classification of dog breeds. The benchmark model is created from scratch and the desired accuracy was around 10%. The desired accuracy level provides us a sufficient hint that our model is doing fine. Because a 10% accuracy implies that the model predicts 1 time correctly out of 133 breeds. I played around with the values to arrive the highest accuracy. The required accuracy for the model was around 10%, I obtained the accuracy near to 15%.

8. Data Preprocessing

Since, this project implements Convolutional Neural Network NN using Pytorch library, while Pytorch may provide better results when we have input as tensor (which is also referred as array). Therefore, the images were first converted to tensor to get them ready for further analysis. Since we are working with color images, each image has three channels. Likewise, since we are processing a single image (or sample), the returned tensor will always have shape

9. Implementation

The process of implementation of the model is as follows:

The cnn model has standard parameters to build a CNN architecture, that can also be customized accrodingly. We can also note two important elements from the standard cnn architecture i.e., CONV and POOL. Where CONV are convolutional layers while POOL are pooling for translational invariance.

I used four convolutional layers in the model by using Conv2d and kernel_size = (3, 3), stride=1, padding=1) for the first two layers while kernel_size = (2, 2), stride=1, padding=1) for the second and third layers.

In our model above, each Conv2D defines CONV convolutional layer to the model, and each MaxPooling2D POOL applies max pooling to the convolutional layer. A 2x2 max pooling layer is applied to the model. Maxpooling layer here will help to reduce the filter dimensionality. The filter size as suggested by the standard practice was set from 3 to 32 and finally to 128. ReLU (Rectified Linear Unit) Activation Function is applied here. Because, the ReLU is the most used activation function in the world right now I followed the intrsuctions provided in the course and specify each convolutional layer and max pooling accordingly.

I used the MaxPooling2D type that is most popular and common in building CNNs. As we can note, as we increase the layers in the model the model becomes more powerfull in capturing the minor details of the input.

The convolutional layer takes images as input were we can also set number of filters, kernal size, padding, activation and input sapes.

The bottom part of the model includes a Flatten (flatten all inputs) layer followed by dropout. The dropout ration was selected to 20%. Finally, I specified the dropout, where dropout helps avoid overfitting the model.

The major issue I faced with number of epochs and layers and time it was difficult task to arrive at the desired accuracy.

10. Refinement

As described above the objective is to classify a dog breed. We selected to apply CNN Convolutional Neural Network for that task. Keeping in mind the objective of the project the appropriate model for that purpose is CNN because these models are well known for analyzing and classifying images-based data. In this project we started from detecting human face, dog face and built CNN model from scratch. You can access the complete project’s notebook here.

I'll move over to the model that provides us best results by utilizing transfer learning approach. The reason of applying CNN with transfer learning is to improve the performance of the model. Because we have obtained the accuracy around 14% in the previous model .

The assigned target was to achieve the accuracy of at least 60% on the test set. For that purpose, I used VGG pretrained model and the obtained model architecture is as follows (detailed model architecture is explained in the notebook):

After, doing the necessary steps such as compilation, training, loading and testing. We obtained the test accuracy around 72%, that is in the acceptable range (greater than 60%).

Then I created my own algorithm in order to predict the image and the code is presented in my github repo:

11. Model Evaluation and Validation

In this step I evaluated my model on the basis of accuracy matric specified above. I find encouraging results by arriving at the accuracy of around 72%. Since, our target was to achieve at least 60% accuracy, keeping in view the complex problem (discussed in the overview) I think this accuracy level is suitable enough to rely on this model.

12. Justification

The output of the model is better than what I expected given the data because it predicted all of the images of dogs correctly. In addition to the saved images, the model also predicted two human images correctly. Moreover, I supplied an abstract image and the model successfully classified that image as neither dog nor human. The CNN model with transfer learning performs much better since it provides an accuracy of nearly 72%, while the benchmark model relied on the accuracy of around 15%. Therefore, the CNN with transfer learning is a great winner here.

Results

In the previous step I developed an algorithm to predict whether the image (input) contains a dog, a human or neither. Possible outcomes of this algorithm are:

- If a dog is detected in the image, it returns its corresponding breed
- If a human is detected in the image, it returns most resembling dog breed.
- In third situation if neither is detected it may return an error message.

In this step I tested the algorithm and the output is as follows:

Figures:1

```
display_image(img_path)
run_app(img_path)
```

```
hi, Doggy!
You belong to a dog breed of .....
Curly-coated retriever
```

Figure:2

```
In [43]: img_path = "images/Labrador_retriever_06457.jpg"
         display_image(img_path)
         run_app(img_path)
```

```
hi, Doggy!
You belong to a dog breed of .....
Great pyrenees
```

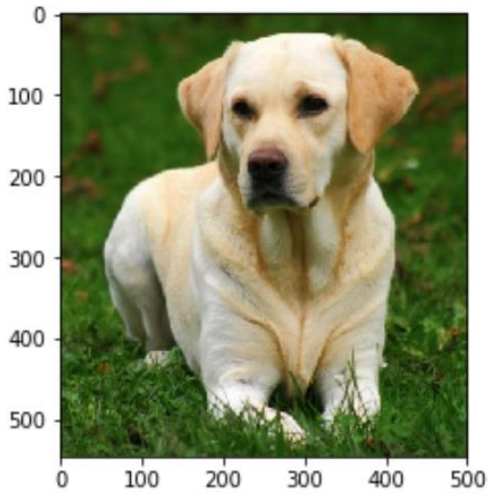


Figure:3

```
In [48]: img_path = "images/bush.jpeg"
         display_image(img_path)
         run_app(img_path)
```

```
hi, Gentleman!
You look like a.....
American foxhound
```

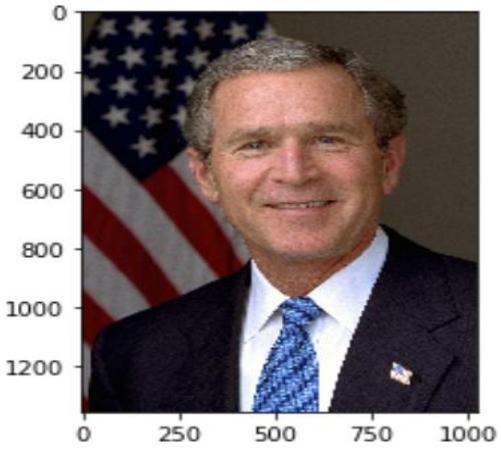
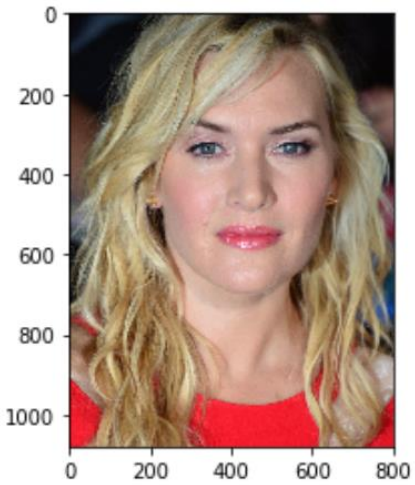


Figure:4

```
In [53]: img_path = "images/kat.jpeg"
         display_image(img_path)
         run_app(img_path)
```

```
hi, Gentleman!
You look like a.....
Nova scotia duck tolling retriever
```



13. Conclusion

The objective of the project was to build a model that correctly identifies the dog breed with an accuracy of over 60%. In order to achieve that objective, I used Convolutional Neural Networks CNN with transfer learning approach I used vgg19 pretrained model in the pytorch. I obtained encouraging results, since my model's accuracy was around 72% and it predicted all the images that I provided as input. As compared to the previous model where we find an accuracy relatively low (15%), the last model provides higher accuracy due to transfer learning techniques.

14. Improvements

There's always room for improvement, we cannot say that our model is perfect. The model can be further improved as suggested in the answers to Question 6.

The output of the model is better than what I expected given the data because it predicted all of the images of dogs correctly.

i. We can always improves the performance of our model by increasing the amount of data.

ii. It might also be a good idea to jointly use neural networks like Neural network ensemble.

iii. In order to improve the performance we could also use augmentation techniques along with playing around with some additional improvement mechanisms such as hyper parameters and gradient descendant optimizer as well.

To know more about the project and implemented model, you can visit my GitHub here .

References

Hu, Wei, et al. “Deep convolutional neural networks for hyperspectral image classification.” *Journal of Sensors* 2015 (2015).

Lundervold, Alexander Selvikvåg, and Arvid Lundervold. "An overview of deep learning in medical imaging focusing on MRI." *Zeitschrift für Medizinische Physik* 29.2 (2019): 102-127.

Maggiori, Emmanuel, et al. “Convolutional neural networks for large-scale remote-sensing image classification.” *IEEE Transactions on Geoscience and Remote Sensing* 55.2 (2016): 645-657.

Rawat, Waseem, and Zenghui Wang. “Deep convolutional neural networks for image classification: A comprehensive review.” *Neural computation* 29.9 (2017): 2352-2449.