10

Dog Breed Classification Using CNN

Sandra Varghese* **and Remya S**

Computer Science and Engineering Department, Saintgits College of Engineering Kottukulam Hills, Pathamuttom, Kottayam, India

Abstract
Dogs are among the foremost common livestock. Due to an outsized number of dogs, there are several issues like social control, decreasing outbreaks like Rabies, vaccination control, and legal ownership. At present, there are over 180 dog breeds. Each dog breed has specific characteristics and health conditions. In order to supply appropriate treatments and training, it is essential to spot individuals and their breeds. This paper presents the classification methods for dogs. It relies on a project that builds a CNN (Convolutional Neural Network) to classify totally different dog breeds. If the image of a dog is found, this algorithm would notice the estimate of the breed. The given system employs innovative strategies in deep learning, also convolutional neural networks and transfer learning. The projected network is prepared to achieve associate accuracy of 93.53% and 90.86%, for two totally different datasets. The result shows that our retrained CNN model performs better in classifying dog breeds.

Keywords: CNN, deep learning, machine learning, artificial neural network, transfer learning

10.1 Introduction

In the US, about 105 million people own one or more dogs, and dog ownership has risen by 29% within the past decade [1]. In Canada, nearly 41% of households have a minimum of one dog [13]. A study conducted by Weiss *et al.* [3], acknowledged that 14% of dogs were lost within the past five

**Corresponding author*: sandra.css1921@saintgits.org

Shibin David, R. S. Anand, V. Jeyakrishnan, and M. Niranjanamurthy (eds.) Security Issues and Privacy Concerns in Industry 4.0 Applications, (195–206) © 2021 Scrivener Publishing LLC

years within the US while 7% of them never returned to their owners. The normal means to spot and locate lost dogs are ID collar, microchip, tattooing and GPS tag [15]. The primary three methods include the knowledge of the dog's name and owner's telephone number which can be helpful when shelters find the lost dogs. GPS tags enable owners to locate their lost dogs directly. Nevertheless, the semi-permanent methods like GPS tag and ID collar are susceptible to being lost or damaged. Meanwhile, the permanent means like microchip and tattooing are not prevalent due to their expensive prices [15]. In rural or remote areas of the northern part of Canada, there is less familiarity with tattoos or dog tags, therefore making it difficult to use the identity of the dog to establish ownership. For veterinarians, this presents an obstacle in retrieving the dog's medical records and, accordingly, prevents adequate healthcare. The concept of e-health for animals is among the main driving forces for promoting the creation of electronic medical records for pets [5]. For every record, a photograph of the pet is taken and may be used to verify/identify the identity of every individual pet using image processing techniques. For instance, an individual who has found a dog in a remote area can take its photo employing a multimedia device and send it to a regional veterinary database for identification. Photos of a dog's head or face are often enough to identify that dog, which is analogous to identifying an individual using the person's face. This is often a fine-grained classification which refers to classifying objects sharing similar visual features and belonging to an equivalent basic-level class [16]. Dog owners or veterinarians are more likely to require various pictures of dogs and store these image data for further use like helping to find lost dogs. There also are many apps available within the Google Play Store to seek out lost dogs. As an example, "Pets finder" only requires a report including the situation and an image of the lost or found pet; the report is uploaded to the cloud database where owners can look for their lost pets among all the pet listings within the vicinity. Multimedia image processing and recognition are often applied to not only identify the dog's face but also breed, height and other soft biometric attributes. During this study, we investigate this approach, specifically, how classifying the breed helps improve dog identification by appearance (face). We apply contemporary machine learning like deep neural networks, also the technique called transfer learning [17] which uses information developed for one task (breed identification) to affect another task (dog face identification).

10.2 Related Work

A substantial amount of research has gone into fine-grained classification problems, the bulk of which has focused on increasing the performance or accuracy of the classification by various approaches. Of these works, some have approached the problem similar to [2], where image processing is employed at the beginning of the process [2]. The provided annotations of the Stanford Dogs dataset, which had locations of bounding boxes that outlined the useful information of each image, was used. Specifically, this meant that the information pertaining to the dog could be found inside these boxes. Using this, the images were all cropped to the bounding boxes, and [2] removed any resulting images smaller than 256X256.

Once this image pre-processing was completed, [2] used LeNet and GoogLeNet architectures. However, [2] noted that transfer learning was not used [3]. Researched fine-grained classification of dog breeds using part localization. The method of [3] employed a number of computer vision topics, focusing on the use of dog faces to improve accuracy in classifying the various breeds. [3] found improved performance through the use of their method; however, this method requires a good number of steps and [4] aimed to reduce the complexity of this sort of approach [4]. Used the Grassmann manifold to represent the geometry of dog breeds. Specifically, [4] focused on the geometry of dog faces and found that their method performed on par with other, more complex, approaches.

The main takeaway here is that the goal of all these works was to enhance performance on a fine-grained classification problem. This is an important issue, as discussed in [2], [3], and [4] as it poses a number of problems. Each related work discussed here approaches the problem in a slightly different way; however, the main ideas behind their approaches are all the same. Each work focused on reducing the amount of information analysed, hoping to reduce the amount to just what is important. Focusing on just what's expected provides useful information in classifying dog breeds. However, we have elected to go a different direction with our research. We have decided not to look into a way for optimizing the networks used, instead, we will be investigating what networks find important in classifying the breeds in the Stanford Dogs Dataset.

To examine the performance of recognizing different dog breeds using assorted images, Khosla *et al*. created a dataset of various dog breeds called Stanford Dogs Dataset [7]. Using Stanford Dogs Dataset, Sermanet *et al*. proposed the use of an attention-based model for breed classification which achieved an accuracy of 76.80% [8]. Another model using depth-wise separable convolutions was proposed in [9] and yielded 83.30% accuracy. Similarly, another dataset, Columbia Dogs Dataset, containing images of different dog breeds, was created by Liu *et al.* [10]. In [10], a breed classification rate of 67.00% can be achieved by using a combination of grey-scale SIFT descriptors and colour histogram features to train an SVM. Furthermore, [11] uses the labelled landmark metadata to improve the accuracy to 96.50% by incorporating Grassmann manifold to help distinguish the different dog breeds using their face geometries. Pet breed classification using both shape and texture was proposed in [12]. A deformable part model was used in [12] to detect the shape and a bag-of-words model to capture the appearance.

For a dataset that includes 37 breeds of cats and dogs, an accuracy of 69.00% was reported. Paper [13] was one of the first to describe dog face identification. It used Fisher Linear Projection and Preservation with One-Shot Similarity for matching and reported 96.87% rank 4 accuracy. The reported dataset is not publicly available, thus making it difficult to perform a proper comparison. Another research [14] was conducted in 2016 aiming at cat face identification by exploiting visual information of cat noses. They designed a representative dictionary with data locality constraint based on a dataset containing 700 cat nose images from 70 cats. This method reached an accuracy of 91.20%. Moreira *et al*. evaluate the viability of using existing human face recognition methods (Eigen Faces, Fisher Faces, LBPH, and a Sparse method) as well as deep learning techniques such as Convolutional Neural Networks (BARK and WOOF) for dog recognition [15]. Using a dataset of two different breeds of dogs, huskies and pugs, Moreira *et al.* show that based on using the WOOF model, an accuracy of 75.14% and 54.38% is obtained for huskies and pugs, respectively.

10.3 Methodology

Even though lot of algorithms are suggests, there are still some problems which make it challenging to realize a satisfying accuracy. We have implemented a CNN model to reinforce the accuracy of dog breed classification [19]. Steps taken in our experiment are,

Step 1: Import Datasets
Step 2: Detect Dogs
Step 3: Create a CNN to Classify Dog Breeds
Step 4: Train a CNN to Classify Dog Breeds (from scratch)
Step 5: Train a CNN to Classify Dog Breeds (via transfer learning)

Step 1: Import Datasets
Our initiative involves loading the datasets that are divided into train, validation and test folders. We would wish to classify dogs into 133 different breeds that are found in our training dataset. We will be using 6,680 dog images to show the models that we will be using. We will be using 835 validation images to fine-tune our parameters, and testing the last word model's accuracy on 836 test images. The test dataset contains images which the model has not seen before.

The datasets contains.

- *Dog Images* - the dog images given are available within the repository within the pictures directory further organized into the train, valid and test subfolders
- *Haarcascades* - ML-based method where a cascade function is trained from tons of positive and negative images, and will detect objects in other images. Therefore the expectation is that a picture with the frontal features clearly defined is required.
- *Test Images* - a folder with certain test images is added to be ready to check the effectiveness of the algorithm
- Pre-computed features for networks currently available in Keras (i.e., VGG19, InceptionV3, and Xception) made available from S3
- Any other downloads to make sure of the graceful running of the notebook are available within the repository. As the next step load the libraries.
- The libraries required are often categorized as follows:
- *Utility libraries* - random (for random seeding), time it (to calculate execution time), os, pathlib, glob (for folder and path operations), tqdm (for execution progress), sklearn (for loading datasets), requests and io (load files from the web)
- *Image processing* - OpenCV (cv2), PIL • Keras and Fastai for creating CNN • Matplotlib for viewing plots/images and Numpy for tensor processing [18].

Step 2: Detect Dogs
Here we use resnet50 pre-trained model in Keras to detect dogs in images. The primary step here is to convert RGB coding of .jpg images into BGR then normalize the three channels supported mean and variance obtained from the large ImageNet database [20]. Luckily, this is often done by the preprocess_input within the applications.resnet50 module in Keras.

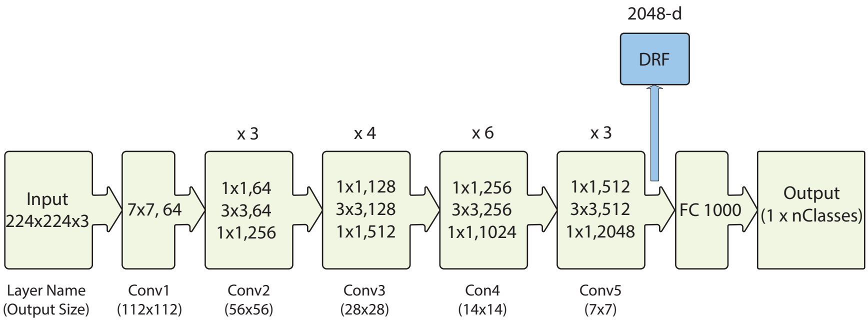

Fig. 10.1 resnet50 model architecture.

Using the resnet50 model we imported from keras.applications library, we will classify the pictures into labels. Within the resnet library any label coded as from 151 to 268 is in effect a "dog". Fig. 10.1 shows the resnet50 model architecture [6]. The function below will give us a real Boolean value if this is the case and False otherwise.

Step 3: Create a CNN to classify dog breeds

We will be using below the bottom Sequential model and designing our CNN networks. We have even used three Convolutional Layers, each followed by a MaxPooling2D to reduce complexity of the stacked model. At the end, we will be using global_average_pooling to convert each feature map within the Max Pooling Layer into a scalar [21]. Fig. 10.2 shows the CNN model architecture.

Step 4: Train a CNN to classify dog breeds (from scratch)

We start training the model we created, and we see that our validation loss constantly lowers, which our accuracy lowers for five epochs, signalling

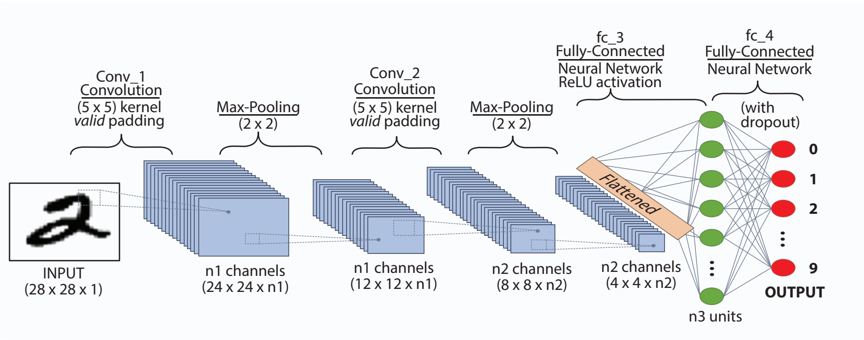

Fig. 10.2 CNN model architecture.

our model is learning. At 20 epochs, it is possible to reach around 4.5% accuracy. Once we run 250 epochs on this dataset, we are ready to reach 40+% accuracy, which does take quite a while even with strong GPU support, which is why we will be using transfer learning as a next step [24].

Step 5: Train a CNN to classify dog breeds (via transfer learning)
Next, we will be using pre-extracted "bottleneck features" which are an output of a pertained library applied on our train, test and validation datasets [22].

We will be using resnet50 library output as an input to our model and train our model using these bottleneck features. We make an easy model by adding a worldwide Average Pooling layer that summarizes each of the previous feature maps into a scalar. The dense layer creates 133 different outputs, one for every needed label. Then, a softmax activation converts each of those into a probability [23].

We then test our algorithm live to ascertain whether it correctly predicts the breed of the dog within the loaded image path [25]. We get a Welsh springer spaniel image loaded into the model, and the neural network correctly classifies the sample dog image that we used [26].

10.4 Results and Discussions

10.4.1 Training

At 20 epochs, it is possible to reach around 4.5% accuracy. Once we run 250 epochs on this dataset, we are ready to reach 40%+ accuracy, which does take quite a while even with strong GPU support, which is why we'll be using transfer learning as a next step.

We are ready to achieve a training accuracy of 93.53% and a validation accuracy of 90.89% using CNN algorithm. Our algorithm is additionally performing excellently on the unseen testing image dataset from Kaggle. (However, thanks to the rationale that labels are not provided for the test assail Kaggle, we are not ready to check the general test accuracy.) The ultimate results seem pretty accurate; we will see that the dog breed prediction for dogs are all correct.

10.4.2 Testing

For 20 epoch, model evaluates a test accuracy of 90.86%. The training accuracy 93.53% is close enough to the test accuracy. Therefore the model can classify images correctly.

Transfer learning works far better than the CNN model. This is mainly because the model from transfer learning is trained by an outsized amount of knowledge, therefore the architecture already understood what features are most representative for an image. This makes the classification process far easier, and we do not have to sacrifice accuracy, albeit we do not have an outsized amount of knowledge. Dog Breed Classification using CNN model was achieved using Keras Library. The model used 20 epoch, i.e., all training dataset skilled aerial and back propagation through the neural network 10 times.

Fig. 10.3 plots the model accuracy and epoch. It is evident that model accuracy increases with the increase in epoch, i.e., here model accuracy is directly proportional to the epoch. Fig. 10.4 depicts the Model loss vs. epoch.

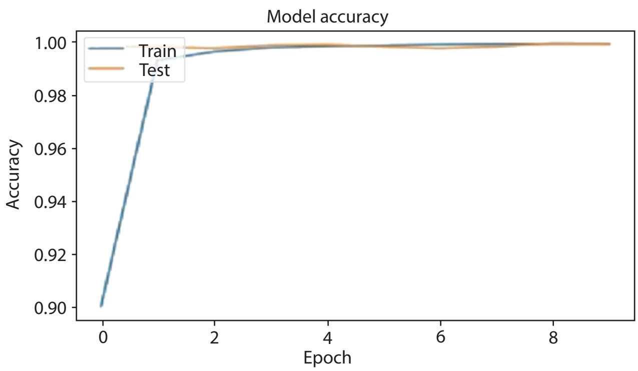

Fig. 10.3 Model accuracy vs. epoch.

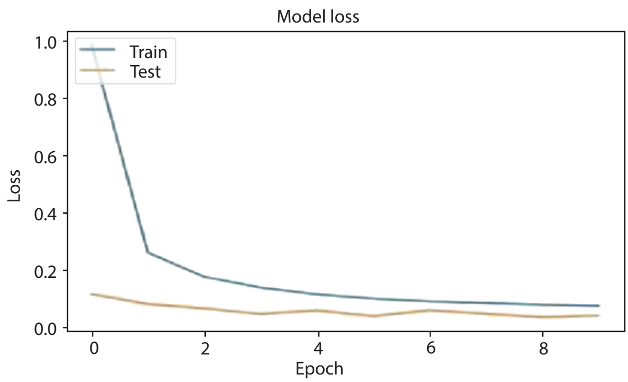

Fig. 10.4 Model loss vs. epoch.

10.5 Conclusions

End-to-end problem solution: this paper offered an answer which may take the image in and then return the breed of the dog. The most interesting aspect of this project is that the magic of transfer learning enables us to produce a better result, albeit we do not have enough data, but still we will train the pre-trained model for our purpose; this is often the most beautiful thing about transfer learning. The convolutional neural networks are often further developed by the following: Generative Adversarial Nets (GAN) [27] to increase the training dataset, using other loss function like centre loss [28], training other convolutional neural network architectures, expanding the dataset with other popular dog breeds, using detectors for locating multiple dogs on a picture and optimizing the server and mobile classification.

The possible improvements on this algorithm might be to urge more data to coach the dog breed classification model. An attempt could be made to tune the model more by using different transfer learning models, and to add more layers to the prevailing architecture. Accuracy is often further enhanced by data augmentation. Data augmentation enables the network to differentiate the features regardless of the orientation and scale. Clearly, building a convolutional neural network using transfer learning yielded much better accuracy than building it from scratch.

There are a couple of breeds that are virtually identical and are sub-breeds. There is also an opportunity of some images being either blurred or having an excessive amount of noise. There is also an opportunity of enhancing the standard by additional image manipulation.

Following the above areas, we could increase the testing accuracy of the model. A simple web application in Flask might be built to leverage the model to predict breeds through user-input images.

References

1. Krizhevsky, Alex, IlyaSutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In *Advances in neural information processing systems*, pp. 1097-1105. 2012.
2. "Latest Canadian pet population figures released," *Tech. Rep.*, 2017.
3. E. Weiss, M. Slater, and L. Lord, "Frequency of lost dogs and cats in the united states and the methods used to locate them," *Animals*, vol. 2, no. 2, pp. 301–315, 2012.
4. S. Kumar and S. K. Singh, "Biometric recognition for pet animal," *Journal of Software Engineering and Applications*, vol. 7, no. 5, pp. 470–482, 2014.

5. B. Yao, G. Bradski, and L. Fei-Fei, "A codebook-free and annotation free approach for fine-grained image categorization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3466–3473.
6. L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, 2010, pp. 242–264.
7. D. Hsu, "Using convolutional neural networks to classify dog breeds," CS231n: Convolutional Neural Networks for Visual Recognition [course webpage]. 2015. [Online]. Available: http://cs231n.stanford.edu/reports/2015/pdfs/fcdh_FinalReport.pdf.
8. J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur, "Dog breed classification using part localization," in *Proc. European Conference on Computer Vision [ECCV]*, 2012, pp. 172-185.
9. X. Wang, V. Ly, S. Sorensen, and C. Kambhamettu, "Dog breed classification via landmarks," In *Proc. 2014 IEEE International Conference on Image Processing*, 2015, pp. 5237–5241.
10. Liu, Jiongxin, Angjoo Kanazawa, David Jacobs, and Peter Belhumeur. "Dog breed classification using part localization." In *European conference on computer vision*, pp. 172-185. Springer, Berlin, Heidelberg, 2012.
11. Zeng, Yujun, XinXu, Yuqiang Fang, and Kun Zhao. "Traffic sign recognition using extreme learning classifier with deep convolutional features." In *The 2015 international conference on intelligence science and big data engineering (IScIDE 2015)*, Suzhou, China, vol. 9242, pp. 272-280. 2015.
12. Saravanan, C. "Color image to grayscale image conversion." In *2010 Second International Conference on Computer Engineering and Applications*, vol. 2, pp. 196199. IEEE, 2010.*FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.
13. Wang, Yu, Qian Chen, and Baeomin Zhang. "Image enhancement based on equal area dualistic sub-image histogram equalization method." *IEEE Transactions on Consumer Electronics* 45, no. 1 (1999): 68-75.
14. Najafabadi, Maryam M., FlavioVillanustre, Taghi M. Khoshgoftaar, NaeemSeliya, Randall Wald, and EdinMuharemagic. "Deep learning applications and challenges in big data analytics." *Journal of Big Data* 2, no. 1 (2015): 1.
15. Gulli, Antonio, and Sujit Pal. *Deep learning with Keras*. Packt Publishing Ltd, 2017.
16. Géron, Aurélien. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2019.
17. "Pet population and ownership trends in the U.S.: Dogs, cats, and other pets, 2nd edition," *Tech. Rep.*, 2017.
18. A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization: Stanford dogs," in *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2011, pp. 1–2.

19. P. Sermanet, A. Frome, and E. Real, "Attention for fine-grained categorization," *International Conference on Learning Representation*, pp. 1–11, 2015.
20. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang,T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.
21. J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur, "Dog breed classification using part localization," in *European Conference on Computer Vision*, 2012, pp. 172–185.
22. X. Wang, V. Ly, S. Sorensen, and C. Kambhamettu, "Dog breed classification via landmarks," in *IEEE International Conference on Image Processing*, 2014, pp. 5237–5241.
23. Y.-C. Chen, S. C. Hidayati, W.-H. Cheng, M.-C. Hu, and K.-L. Hua, Locality constrained sparse representation for cat recognition," in *International Conference on Multimedia Modeling*, 2016, pp. 140–151.
24. T. P. Moreira, M. L. Perez, R. de Oliveira Werneck, and E. Valle, "Where is my puppy? retrieving lost dogs by facial features," *Multimedia Tools and Applications*, vol. 76, no. 14, pp. 15325–15340, 2017.
25. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards realtime object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.
26. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representation*, pp. 1–14, 2015.
27. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
28. Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 499–515.