Mixed and Pure Breed Dog Classifier

Luke Trinity and Nat Shenton
CS 254

December 7, 2019

1 Introduction

Online sources of high resolution images including Google, Image-net [1], and Flickr are providing new opportunities to explore applications of image categorization using computer vision. One interesting application in the field of fine-grained image classification is identification of dog breed. Dogs faces are considered to be highly differentiable for each breed, which makes classification an approachable problem. However, the work is challenging due to the varying combinations of facial characteristics and color patterns between species, as well as intra-class variation within a single species. Another aspect of the problem that makes it more complicated is the presence of humans and man-made backgrounds, which are not as common in other animal datasets. In addition, many attempts to classify dog breeds fail to corroborate the results with actual DNA evidence. Our approach will be to utilize an existing dog breed image classification dataset [2], in combination with a novel dataset that includes paired DNA test results and photos [3], to classify non-pure bred dog breeds or mixed breed.

2 Problem Definition and Algorithm

2.1 Task Definition

The task is to take dog images where each image is labeled either mixed or pure breed and use them to train an algorithm to classify an unseen photo based on whether the dog is mixed or pure. An important part of training our neural network is to normalize the input images. Each input image is defined as a matrix of $250 \times 250 \times 3$. The output of this algorithm, will either be a mixed or pure bred dog. This is an interesting problem as it could provide dog owners an easy way to classify if their dog is a pure bred or mixed, and combines multiple methods of machine learning. It could be an exciting way to start a conversation and help people learn about different breeds of dogs, and how they are mixed. The final goal for this task would allow for a dog owner to upload a picture of their dog to a website or app and then the machine learning algorithm will classify their dog as the pure bred breed or the top three possible breeds.

2.2 Dataset

The Stanford Dogs Dataset has 20,580 images each classified into one of 120 classes of dog breed. [2] Each image also has an associated bounding box identifying the location of the dog within the photo. The mixed-breed DNA identified dog photo dataset from Voith et al. [3] has 20 images, each paired with DNA percentage breakdown of dog breed. The final mixed breed data set a combination of the top 25 mixed dog breeds from dogfinder.com, and crowdsourcing bring the total up to **number of mixed dogs** pictures. There was no way to actually identify if the dog was actually mixed or not so we assume that the submission of the mixed breed dog is correct in the data. The final dataset was obtained by a random sampling of pure bred dogs from the Standard dataset equal to the length of the amount of dogs form the Stanford Dogs Dataset. This created a dataset of total number of dogs. This created a dataset of **number of photos**.

2.3 Algorithm Definition

First, we utilize the Resnet50 convolutional neural network [11] as it achieves roughly 80% accuracy on the Standford dog dataset. This pretrained neural network gives a classification accuracy high enough that it should classify an image of a purebred dog correctly. This pretrained convolutional neural network is trained on Google's ImageNet which contains the Stanford Dogs dataset. By feeding images from the mixed dog breed dataset into this pretrained neural network, the network gives the output as the probability of each dog breed. The output contains the probabilities of 1000 labels, and the algroithm will only take the labels with the top 4 highest probabilities. From here principal component analysis is used to take the four prediction outputs from the pretrained convolutional neural network down to two features. With these two features as the input classes, a sweep of three classification methods were used SVM, random forrest, and a neural network.

For the SVM classifier two techniques are used. The first being a random sweep over a linear kernel with a C values randomized over an exponential distribution starting at 1000, and γ values randomized over an exponential distribution starting at 2. Other SVM algorithm has the same C and γ sweeps but also randomly chooses a Radial Basis Function kernel or a polynomial kernel. The random forest algorithm looks at a forest of 10 trees. Lastly the neural network uses an architecture of 3 hidden fully connected layers with the ReLu activation function. The first layer contains 48 nodes, the second 24 nodes, and the third 4 nodes. The output layer contains one node with a sigmoid activation function. The model was then compiled with the Adam gradient descent optimizer, and binary cross-entropy as the loss function. The neural network was trained over 300 epochs.

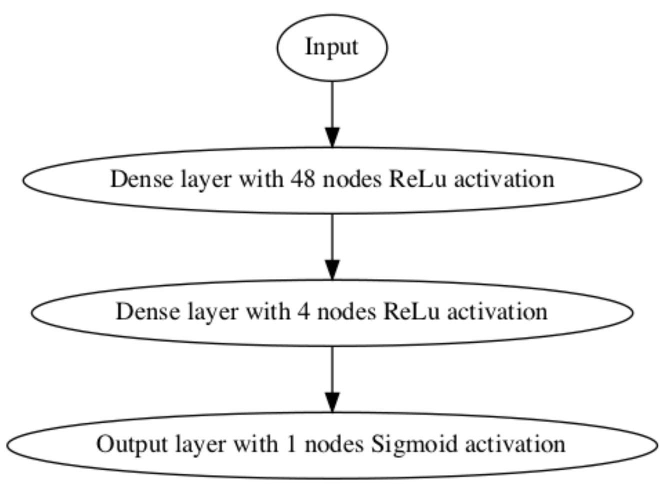

Figure 1: Initial Neural Network Structure

Cross validation is run on each model type to fit the optimal algorithm in terms of accuracy score. These four types of models were each chosen to run with random parameters 250 times each to find the optimal model from each model type. The random forrest and the nueral network did have the same parameters but were always initialized the same. Of these four the model with the highest accuracy score in the validation stage is chosen to be the final model.

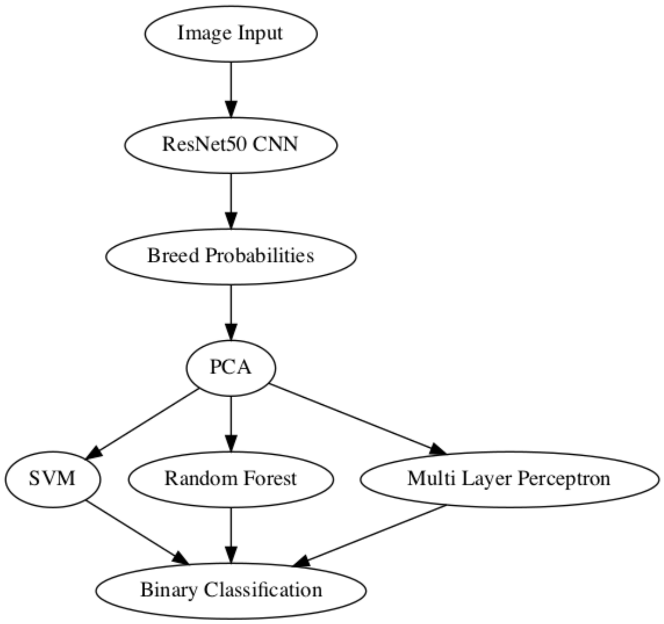

Figure 2: Algorithm Structure including testing suite

3 Experimental Evaluation

3.1 Methodology

By taking a dataset of **number of images**. A testing suite was used to test each algorithm. The testing suite randomly selects an algorithm SVM, Random Forrest, and Multi Layer Perceptron Neural Network. For the SVM and Random Forrest it randomly chooses the parameters as stated above. The two PCA features are then used to for each algorithm to find the best non linear decision boundary. Below in figure 3 is the dataset after PCA.

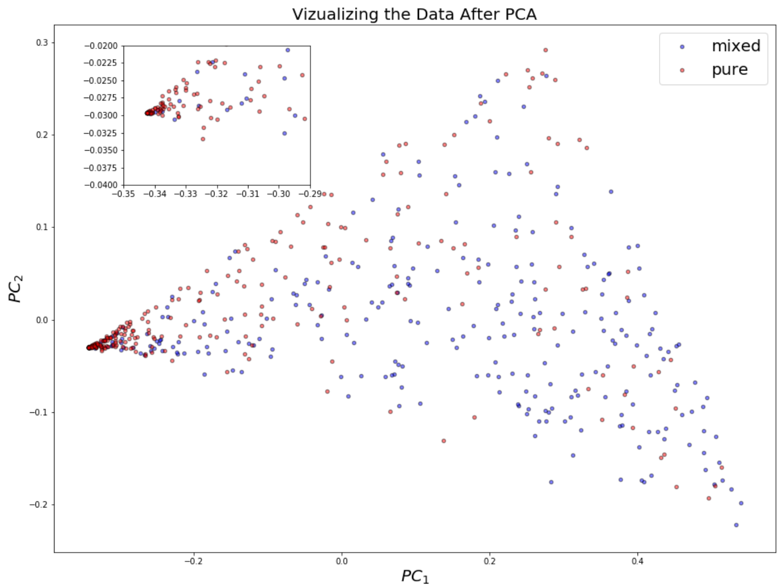

Figure 3: The distribution of the data after using principal component analysis to reduce the breed probabilities from 4 features to 2. Red is purebred while blue is mixed. The inset plot shows the area where the data in clustered.

This data will require a non linear hence why the testing suite is need to sweep over many models to find the one that separates the data the best. The testing suite will also run over many testing and training splits as the data set is small so the many splits are necessary to account of variability in the data

A second testing suite is used to test multiple neural network architectures to optimize the dataset.

3.2 Results

Here are the results of the initial testing suite. Figure 4 shows the distribution of accuracy scores. Here this shows the initial testing suite results with each model run 250 times. The SVM are using random search cross validation what the only parameters changed in the testing.

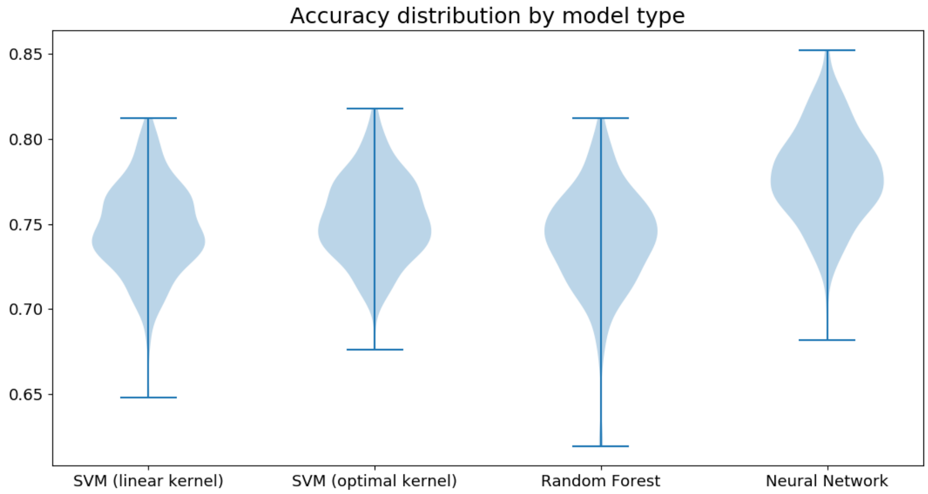

Figure 4: The violin plot shows the distribution of testing accuracy of each model type from the testing sweep.

Figure 4 shows that SVM with linear kernel, SVM with a non-linear kernel, and random forests perform about the same, but the neural network with the architecture shown in figure 1 gives the best accuracy. Figure 5 gives the $\text{Accuracy}_{\text{Testing}} - \text{Accuracy}_{\text{Training}}$ for each model. This gives a representation of the overfitting and under-fitting the models could be experiencing. Here it seems that the neural network is overfitting the data with the mostly negative values.

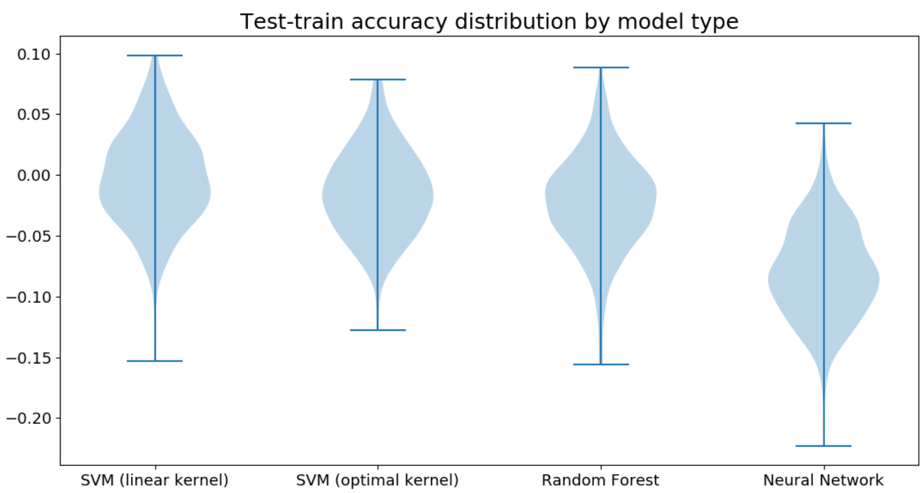

Figure 5: The violin plot shows the distribution of the $\text{Accuracy}_{\text{Testing}} - \text{Accuracy}_{\text{Training}}$ for each model.

3.3 Discussion

Interestingly, the accuracy increased slightly from $N = 6$ to $N = 7$. This may not be significant, but in the future we can average over multiple runs to determine if the trend is consistent. It is also possible that with more breeds to train on, the model could become more accurate because certain breeds are very unique and easy to classify. We believe this is an excellent output, with the potential to scale easily and help us accomplish our intermediate goal of classifying any pure bred dog, before moving on to mixed breeds classification. The algorithm has great potential for improvement, with only the barebones framework currently in place.

4 Related Work

In the initial article utilizing the novel dataset by Khosla et al. [2], the mean accuracy reached 22% when 100 training images were utilized within the SIFT methodology. Liu et al. [4] reached 69% accuracy, with greater success than Khosla et al. [2] due to their partially localized approach. We hope to leverage the bounding boxes and augment the work of Liu et al. [4], possibly improving on their methodology to achieve higher accuracy. Several bounding box approaches are currently being reviewed, although all are endlessly modifiable from simple image segmentation. One is a slight adaptation to a method often employed for training algorithms to images of different shade and dimension called DeepCut. [5] There are several standard variations on the basic DeepCut methodology, which hones in on the nature and dimension of the bounding in a cookie-cutter, traced-out way as opposed to any kind of geometric shape. A very similar methodology is known as GrabCut which does not need to even be a single, continuous cookie-cutter-style line bounded around a desired segment of an image. GrabCut is standardly designed to grab multiple disparate segments of an image at once for a single, desired analysis. In addition to one of these approaches or some modification thereof, we plan to incorporate some aspect of KL Loss to account for localization uncertainty which could greatly assist with interference from background objects such as people, buildings, trees, etc. [6] There are, of course, a number of other modifiable 'standard' or 'familiar' combinations of approaches to deal with bounding and uncertainty to also be explored. Even so, most will lead to an approximately similar result if implemented correctly. Deep/GrabCut is simple and can narrow the selected image segment in a convenient, efficient manner. KL Loss will be further researched and incorporated to deal with expected shading difficulties, image blurring and background/foreground image 'noise'.

5 Code and Dataset

The code can be found in our github repo: `https://github.com/nshenton/CS_251_DogProject`. Please see the Readme in the repo for more information. The Stanford Dogs Dataset can be downloaded from this page: `http://vision.stanford.edu/aditya86/ImageNetDogs/`.

6 Conclusion

Here we have described our barebones methodology and presented intermediate results. Our path forward will involve a team effort of improving the accuracy of our model, and leveraging the technologies of the VACC. We believe our final product will be interesting and a great conversation starter for dog lovers and machine learning enthusiasts alike. Our elavator pitch is simple, wouldn't you like to upload a photo of your dog and see what breeds it could potentially be? Looking past this semester, it would be a long-term goal to incorporate user feedback into the training of the model, i.e. having users rate the predictions for how accurate they are perceived to be.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[2] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization: Stanford dogs," in *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, vol. 2, 2011.

[3] V. L. Voith, E. Ingram, K. Mitsouras, and K. Irizarry, "Comparison of adoption agency breed identification and dna breed identification of dogs," *Journal of Applied Animal Welfare Science*, vol. 12, no. 3, pp. 253–262, 2009.

[4] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur, "Dog breed classification using part localization," in *European conference on computer vision*, pp. 172–185, Springer, 2012.

[5] M. Rajchl, M. C. Lee, O. Oktay, K. Kamnitsas, J. Passerat-Palmbach, W. Bai, M. Damodaram, M. A. Rutherford, J. V. Hajnal, B. Kainz, *et al.*, "Deepcut: Object segmentation from bounding box annotations using convolutional neural networks," *IEEE transactions on medical imaging*, vol. 36, no. 2, pp. 674–683, 2016.

[6] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2888–2897, 2019.

[7] Google, "Google ai," 2019.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[10] F. Chollet *et al.*, "Keras." `https://keras.io`, 2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.