Automatic Dog Breed Identification

Dylan Rhodes
CS231n, Stanford University
dylanr@stanford.edu

Abstract

In this paper, a system for automatically identifying dog breeds via images is explained, implemented, and evaluated. The system consists of three major stages: facial keypoint localization, facial normalization, and breed classification. The facial keypoint localization algorithm is intended to be my project for CS231A, while the breed identification algorithm is intended to be my project for CS231N. The intermediate phase is insufficient as a project for either class yet complements both. Further, all three stages are crucial to the performance and understanding of the system as a whole, so all three will be at least briefly characterized here with more detail given to the implementation and evaluation of the breed identification network. Both predictive models have ultimately taken the form of convolutional neural networks yet retain several important differences. Results on the breed identification task are compared to those of Liu et al, 2012.

1. Introduction

1.1. Domestication History

There is a great deal of controversy surrounding research into the events surrounding wolves' first domestication into dogs by humans during prehistory. Via genetic analysis of extant populations of both groups, evolutionary biologists have placed this event somewhere in Eastern Europe or Asia between eleven and sixteen thousand years ago [4]. This indicates a date earlier than both the invention of agriculture and the domestication of any other wild species of animal. Thus, our special relationship with *canis lupus familiaris* extends back farther in time than almost any other as well as our organized societies and technologies.

The chronological depth of this relationship is reflected in the astounding diversity of domestic dogs alive today. The household canine is the most morphologically and genetically diverse single species of animal on Earth. There are countless breeds of dogs which display hugely varying physical and psychological traits ranging through color, shape, size, and behavior.

1.2. Breed Identification Problem

This great variety poses a significant problem to those who would be interested in acquiring a new canine companion, however. Walking down the street or sitting in a coffee shop, one might see a friendly, attractive dog and wonder at its pedigree. In many situations, it is impossible to ask an owner about the breed, and in many cases, the owner themselves will be either unsure or incorrect in their assessment. Unless the dog falls into one of a few very widely known and distinctive breeds such as the golden retriever, Siberian husky, or daschund to name a few, it might prove difficult to identify ones ideal companion without a great deal of research or experience.

1.3. Dataset

In a step towards partially rectifying this issue, in 2012, Columbia researcher Jiongxin Liu and his lab released a richly annotated dataset of canine images organized by breed as the Columbia Dogs with Parts Dataset to the *European Conference on Computer Vision* [5]. The collection spans nearly eighty-four hundred images selected from ImageNet, Flickr, and Google Image search. The subjects of these images are fairly evenly split between one hundred thirty-three official breeds recognized by the American Kennel Club. This set includes familiar faces, such as the Chihuahua and Great Dane alongside lesser known breeds like the Borzoi and Icelandic sheepdog. By means of mechanical turk and careful cross checking, each image has also been annotated with the locations of eight facial keypoints: the eyes, nose, top of the head, and base and tip of the ears. For images in which some of the parts are occluded or lie outside of the frame, a best estimate for their location is provided. This dataset forms the basis for my project, which is the application of convolutional neural networks to the breed identification problem. Figure 1 gives a partial illustration of the breeds described within the dataset.

Figure 1. Breeds contained in the Columbia Dogs with Parts Dataset.

1.4. Training Procedure

All models described in this paper were trained on Amazon EC2 spot instances. Shared g2.2xlarge machines containing NVIDIA GRID K520 GPUs running CUDA 6.5 were rented on an as-needed basis. Most models took between eight and twelve hours to train fully, and the total cost of the setup over the course of the project was around fifteen dollars. Initially, Caffe was used for training the convolutional networks, but ultimately, Theano [1] and Lasagne proved more tractable for rapid iteration on model architectures and analysis via easy-to-read Python.

2. Related Work

As far as I can tell, the Columbia Dogs with Parts collection has not received any researcher's attention since its debut in 2012, possibly due to its relatively small size in the era of big data. Nonetheless, its rich annotations and intriguing subject matter invite novel approaches to an extremely compelling problem. Further, Liu's original paper provides an educational first look at the data.

2.1. Part Localization

In Dog Breed Classification Using Part Localization, Liu and his team present a surprisingly successful localization and classification pipeline of their devising for the breed identification problem [5]. As a preprocessing step, they identify a large facial window with another linear SVM sliding window model evaluated over location, scale, and rotation. Non-maximum suppression produces a small candidate window for the face in which to look for the specific keypoints. Their localization algorithm then focuses on identifying the best candidate positions of the nose and eyes, the most easily identifiable parts, and employs a consensus of models approach between three sliding window SVMs, one for each part, over grayscale SIFT descriptors. The keypoint SVMs are then used to construct a heatmap for the locations of the eyes and nose which is refined via a RANSAC-like procedure in which many labeled exemplar images are fit to the modes of the heatmap and the closest matches pooled to produce a final estimate for the parts' locations. This approach to localization produces lower error than the agreement between individual mechanical turk workers used to produce the labels in the first place.

2.2. Breed Identification

Liu's breed identification algorithm also produces impressive results via a pipeline backed by SVMs. For each of the one hundred thirty-three different breeds, they train a one vs all SVM over grayscale SIFT descriptors centered at the predicted part locations and the midpoints of the lines connecting them as well as a quantized color histogram over the entire facial region. Locations of parts other than the eyes and nose are repeatedly estimated by transforming the closest matches for each breed onto the eyes and nose and merging the score of the SVM for each. Ultimately, this method achieves 67 percent first-guess accuracy and 93 percent top-ten accuracy on the breed identification problem, extremely robust results given the large number of classes, high intra-class variability, and often low inter-class variability inherent to the breed identification problem. They also compare this algorithm to other popular model types and find that it outperforms bag of words, multiple kernel learning, and locally constrained linear coding by a large margin.

3. Facial Keypoint Detection

My breed identification algorithm also heavily relies on accurate facial detection as an initial step. Over the course of the project, two approaches to this localization problem

were implemented, trained, and evaluated. Both are briefly presented here.

3.1. First Approach

As a first attempt, I reimplemented Liu's facial detection algorithm with a few minor modifications, since the breed identification model does not require the exact location of the keypoints, just that of the face as a whole, along with its orientation and scale. First, the average displacement of the nose and eyes relative to the center of the face, here defined as their mode, the orientation of the line connecting the eyes, and the interocular distance were all calculated over the dataset's ground-truth labels. With these measurements, the normalized, geometric displacement of the dogs noses and eyes from the center of their faces could be computed and stored. Next, grayscale SIFT descriptors centered at these locations were collected and scaled by the interocular distance for each of the positive training samples. Negative samples were constructed by randomly sampling a section of the training images outside of the ground-truth facial boxes under a scale and rotation drawn from a normal distribution around the mode of their values in the positive training set. Since it was cheap to generate additional negative samples, the final training set contained 4776 positive, normalized examples of canine faces as well as 13,000 negative examples.

With this data in hand, a binary classifier was trained in the form of a linear SVM. In a process identical to that of training set generation, a test set was created - this time including an equal number of positive and negative samples. The positive samples were scaled and rotation normalized as in training based on their ground-truth keypoint labels. Unfortunately, I found that this classifier achieved only 55 percent accuracy on the binary discrimination task between canine faces and non-faces. It is not totally clear why this model performed worse than that of Liu, as his paper does not provide quantitative results of the generic facial detection problem. It is possible that the models are roughly equivalent, since his evaluation procedure consists of pooling scores for candidate windows to select an area for finer grained analysis in order to generate part locations, whereas mine was evaluated on the binary classification task over normalized samples. Regardless, though, I considered this performance unsatisfactory, especially given its near total failure on non-frontal images and opted to start over with a different method.

3.2. Second Approach

The second approach to facial detection proved much more successful at the task. This procedure consisted of a convolutional neural network directly regressed on the ground-truth part locations and therefore able to predict them directly. A fairly shallow convolutional neural network was trained with a mean squared error loss function on the location of all eight facial keypoints. The output of even the prototype network seemed workable, so I settled on that type of model over competing approaches such as deformable parts models and poselets, which had been suggested during office hours.

Figure 2. Predicted facial keypoint locations (red) and ground truth labels (green) for three dogs of varying pose under the final model.

The keypoint localization network went through several rounds of improvements over the course of the project, culminating in a model which could successfully identify the facial region, orientation, and scale for all images with precision comparable to that of Liu et al. Briefly, dropout, color and contrast jitter, leaky ReLu, and an extended network architecture and training time were all employed to boost the models accuracy, but as this paper is focused on the breed identification network, which made use of some of the same procedures, they will not be discussed at length here. If anyone is interested in more details of this model and its performance, I encourage them to read my CS231A paper. Figure 2 illustrates some examples of the output of the final model. With this estimator tuned and sufficiently accurate, I turned towards the next phase of breed identification.

4. Facial Normalization

Once the facial keypoint locations have been predicted by the first phase of the breed identification pipeline, normalized subimages of a constant size are prepared for the second neural network. This intermediate step is extremely important for the performance of any subsequent analysis, so it is described in detail here. First, the center of the face is estimated as the mode of the midpoint between the eyes and the nose. Next, the slope of the vector from the left eye to the right eye is calculated and the entire image is rotated so that it lies flat. Finally, the interocular distance, the length of the segment between the eyes, is calculated and a box centered at the center of the face with side length four times the interocular distance is cropped from the image. This box is then scaled to a constant size to serve as input to the next phase of the pipeline, breed identification. Before and after examples of this process are included in Figure 3.

5. Breed Identification

Effective breed identification is the heart of my project. It received a brief treatment above in the related work sec-

Figure 3. Eight examples of images before and after facial normalization. Concentrating on the face eliminates a huge amount of noise in the image and allows models to focus on a static, rigid object.

Figure 4. Example images of the Norwich terrier, Cairn terrier, and Australian terrier which display low inter-class variability.

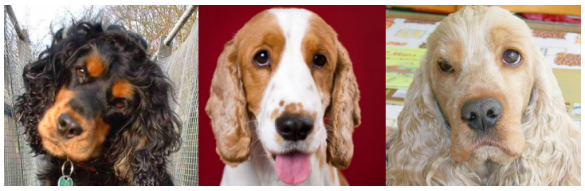

Figure 5. Three members of the English cocker-spaniel breed which demonstrate high intra-class variability.

tion, but here I describe it fully. The problem is one of very fine-grained, specifically 133-way, classification, so even a first-guess accuracy score of one percent is a significant improvement over chance. Historically, computer vision researchers have struggled to effectively distinguish dogs from cats, so it is a fairly difficult task. There are three major issues with the breed identification problem as it pertains to this dataset: low inter-class variance, high intra-class variance, and pose variance.

5.1. Obstacles

5.1.1 Interclass Variance

Low inter-class variance is a phenomenon common within the animal domain. It is well known, for instance, that certain species of birds are more difficult to distinguish from one another than others due to similarity in coloration [3]. There are many breeds of dog which appear more or less identical to the untrained eye, yet to achieve high accuracy, a vision algorithm must be able to pick up on the minor details distinct to each class. In figure 4, there are images of three dog breeds which exhibit this tendency towards similarity.

Figure 6. Three canines pictured in widely varying settings and poses.

5.1.2 Intraclass Variance

The dataset also suffers from high intra-class variance, in which a single breed of canine can take on a variety of appearances naturally. There are several breeds of dog which are best differentiated via the color of their coats, yet many can take on a range of colors and patterns from tawny through black, white, and spotted. The breed identification model must make use of these contradictory signals in order to provide an intelligible estimate of the dogs true identity. Figure 5 illustrates one example of a breed with high intra-class variance.

5.1.3 Pose Variance

Pose variance is a canonical problem within computer vision. In this specific case, it relates mainly to the variety of settings in which dogs have been pictured in the dataset. Canines are extremely deformable and pictured across a large number of tasks including at dog shows, in trucks, at home, and in peoples handbags. This noisy atmosphere makes it extremely difficult for any vision algorithm to separate out meaningful signal for the dogs breed. In figure 6, I have provided some examples of this miscellany of subjects. This problem is arguably the most difficult and fundamental for dog breed identification. Indeed, the performance of a convolutional neural network with my final architecture, hyperparameters, and training epochs yet lacking pose normalization for input images yielded an abysmal 1.4 percent accuracy score.

5.2. Approach

A variety of neural networks were implemented, trained, tuned, and evaluated for the breed identification problem over the course of the project. To maintain the ability to directly compare my results with those of Liu et al, I used his original train/test partition of the dataset into 4776 training images and 3575 testing images. Input images of varying content were cropped, rotated, and resized to fit a sixty-four by sixty-four pixel square across three color channels as described in section 4. Figure 7 gives the quantitative results of each model in a single graph.

As a baseline, I began with a small, fully connected network. This model consisted of a single hidden layer of one thousand units connected to a softmax layer for prediction over the one hundred thirty-three breeds. Surprisingly, this fairly naive network attained a 9.56 percent classification accuracy, which is impressive for such a simple model.

The first convolutional network (Net A in figure 7) improved upon this baseline by a significant margin. A shallow network, it included only three convolutional layers followed by simple rectified linear unit nonlinearities and two by two max pooling. On top of the convolutions, two fully connected layers of one thousand hidden units each as well as a softmax classifier were trained. The model did not include any tricks, bells, or whistles, and can be considered a translation of my dense network into a convolutional form. This network reached saturation in under an hour of training with a final accuracy score of 21.11 percent.

Following these experiments, I decided to expand the network architecture in order to see if its results could be improved. An additional convolution before each pooling layer was included for a total of six and the number of filters available to the original three were increased by a factor of two. Having noticed that the first convolutional model suffered from heavy overfitting with a train/test loss ratio of 0.136, I also added dropout after each pooling layer as well as the first fully connected layer. These modifications pushed the performance of the network slightly higher but also greatly increased the time to convergence to eight hours. The networks final accuracy score was 24.22 percent.

For the final network, I implemented a variety of improvements but concentrated on data augmentation. The size of the training set had been a noticeable hindrance throughout the process, resulting in overfitting and poor performance, so I included a variety of ways to expand it. Color and contrast jitter were included in this model as well as random horizontal mirroring. The input dataset was also expanded by randomly jittering the theta and center of the facial crops of the original images over a normal distribution centered at their true estimates to produce five transformed copies of each training image (the true estimate was also included for a total expansion factor of six). Finally, a leaky ReLu nonlinearity was implemented and swapped into the network following each convolutional layer. This final model achieved an accuracy score of 30.6 percent, a substantial improvement over Net B. The network architecture of this model is given in Table 1.

As an additional experiment, the performance of Net C's architecture over the raw images in the dataset was also evaluated. Thus, the full training and testing images were scaled down to the same size as the randomized crops of Net C and the same network architecture was trained over them. This procedure resulted in a 1.4 percent accuracy score for breed identification, which while nearly double chance performance, remains much lower than even the dense, naive model over preprocessed inputs. This result illustrates the necessity of effective preprocessing before a complicated image classification task such as breed identification.

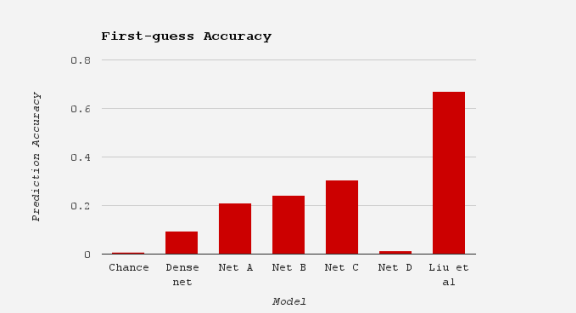

Figure 7. First-guess prediction accuracy for successive iterations of my breed identification model including the final model trained on non-pose-normalized input images.

Layer Type	Filter Size	Volume Size
Input	N/A	(3, 64, 64)
Convolution	(3, 3)	(700, 32, 62, 62)
Convolution	(3, 3)	(700, 64, 60, 60)
Max Pooling	(2, 2)	(700, 64, 30, 30)
Convolution	(3, 3)	(700, 64, 28, 28)
Convolution	(3, 3)	(700, 128, 26, 26)
Max Pooling	(2, 2)	(700, 128, 13, 13)
Convolution	(3, 3)	(700, 256, 11, 11)
Convolution	(3, 3)	(700, 256, 9, 9)
Max Pooling	(2, 2)	(700, 256, 5, 5)
Fully Connected	N/A	(700, 1800)
Fully Connected	N/A	(700, 1000)
Softmax	N/A	(700, 133)

Table 1. Network architecture for final breed identification model.

6. Conclusion

Overall, I am disappointed that I failed to match Liu's reported accuracy on the dataset, although my final results are fairly robust for such a difficult problem. At one point, I incorrectly thought that I had succeeded in surpassing 67 percent accuracy, but in reality, my models do not perform nearly as well as that. Notwithstanding that reality, there are several valuable takeaways from this project as well as remaining avenues of investigation.

6.1. Contributions

There are two main contributions of this project. The first is a conclusive demonstration that the performance of neural networks can be improved via the inclusion of image metadata, such as part annotations. As illustrated by Net

D, the performance of my model is wholly contingent upon successful localization of the facial keypoints followed by effective normalization. This tendency can likely be extended to other image classification tasks; I would expect the performance of any generic image classifier to significantly improve if another is used to locate keypoints and normalize its input beforehand. The second contribution is the verification of an effective augmentation method. Theta and center jittering to produce randomized crops and expand the training data elicited the largest increase in classification accuracy for my model among all of the improvements which were tried. When training a network of this size from scratch with such a small dataset, clearly it is important to generate as much augmented data as possible.

6.2. Future Work

There are several improvements which I would like to make in the future. One intriguing possibility is that of collecting additional breed-labeled dog images from the internet. Since my facial keypoint localization model can accurately bound canine faces, it would be easy to collect unannotated dog images online where they are abundant and use this extended training dataset to further improve the breed identification model. Another interesting idea would be to train the localization model to emit bounding boxes rather than keypoint locations. Simplifying this model could result in greater accuracy, and since the breed identification model only requires scale, location, and orientation of the face, the specific locations of the keypoints are not totally necessary for the prediction task. My final idea is to extract subsets of the data for breeds with either especially low inter-class variability or high intra-class variability to determine how best to fit a model to data with those characteristics. In the final setup, the only real response to variability issues was to increase regularization and try to expand the dataset, but there may be more effective means of teaching a model to finely distinguish very similar classes or group images which are not totally alike.

Moreover, I believe that simpler improvements could easily be made to my final model. Due to my late realization of a bug in my code (see next section), I only ended up with enough time to run my final model for three hundred epochs over the augmented training dataset. If I were to continue running the model for a greater period of time, it probably would have continued improving, as the validation loss was still decreasing when it was terminated. Further, the model likely would have benefited from a stronger dropout ratio which I was unable to employ for the same time restrictions. A final takeaway one might glean from this project is that neural network experiments ought to be set up very carefully and given a great deal of time, space, and cycles.

6.3. Important Clarification

As a final note, I must make an important clarification. Those of you who saw my poster at the presentation on Wednesday will be wondering why my reported final accuracy score has decreased. At that time, I thought that I had surmounted the accuracy of Liu et al with my most recently trained network. However, I later discovered that my accuracy scores had been artificially inflated by a bug in the implementation of the selection of my evaluation set. Basically, after I implemented randomized cropping to augment my training set, I continued sampling evaluation images from this body of randomly cropped images. I did not recognize at the time that closely related crops drawn from the same image could then appear in both the training and evaluation set, which inflated my results. I apologize for misleading attendees of the poster session and have amended my code and results - which are here evaluated on Liu's original test set - accordingly.

References

[1] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[2] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[3] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *Computer Vision–ECCV 2010*, pages 438–451. Springer, 2010.

[4] A. H. Freedman, R. M. Schweizer, I. Gronau, E. Han, D. O.-D. Vecchyo, P. M. Silva, M. Galaverni, Z. Fan, P. Marx, B. Lorente-Galdos, et al. Genome sequencing highlights genes under selection and the dynamic early history of dogs. *arXiv preprint arXiv:1305.7390*, 2013.

[5] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur. Dog breed classification using part localization. In *Computer Vision–ECCV 2012*, pages 172–185. Springer, 2012.

[6] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Cats and dogs. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3498–3505. IEEE, 2012.

[7] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 606–613. IEEE, 2009.