Dog Breed Identification

Whitney LaRow
wlarow@stanford.edu

Brian Mittl
bmittl@stanford.edu

Vijay Singh
vpsingh@stanford.edu

Abstract

This project uses computer vision and machine learning techniques to predict dog breeds from images. First, we identify dog facial keypoints for each image using a convolutional neural network. These keypoints are then used to extract features via SIFT descriptors and color histograms. We then compare a variety of classification algorithms, which use these features to predict the breed of the dog shown in the image. Our best classifier is an SVM with a linear kernel and it predicts the correct dog breed on its first guess 52% of the time; 90% of the time the correct dog breed is in the top 10 predictions.

1. Introduction

1.1. Background

This project hopes to identify dog breeds from images. This is a fine-grained classification problem: all breeds of *Canis lupus familiaris* share similar body features and overall structure, so differentiating between breeds is a difficult problem. Furthermore, there is low inter-breed and high intra-breed variation; in other words, there are relatively few differences between breeds and relatively large differences within breeds, differing in size, shape, and color. In fact, dogs are both the most morphologically and genetically diverse species on Earth. The difficulties of identifying breeds because of diversity are compounded by the stylistic differences of photographs used in the dataset, which features dogs of the same breed in a variety of lightings and positions.

1.2. Motivation

This problem is not only challenging but also its solution is applicable to other fine-grained classification problems. For example, the methods used to solve this problem would also help identify breeds of cats and horses as well as species of birds and plants - or even models of cars. Any set of classes with relatively small variation within it can be solved as a fine-grained classification problem. In the real-world, an identifier like this could be used in biodiversity studies, helping scientists save time and resources when conducting studies about the health and abundance of certain species populations. These studies are crucial for assessing the status of ecosystems, and accuracy during these studies is particularly important because of their influence on policy changes. Breed prediction may also help veterinarians treat breed specific ailments for stray, unidentified dogs that need medical care. Ultimately, we found dogs to be the most interesting class to experiment with due to their immense diversity, loving nature, and abundance in photographs, but we also hope to expand our understanding of the fine-grained classification problem and provide a useful tool for scientists across disciplines.

2. Related Work

Plenty of previous work has been done in the field of fine-grained classification, and we used this literature to develop an understanding of the field [9] [3] [5]. Likewise, there was a fair amount of research that has been done into part localization, which we heavily leverage in our project [7] [4] [1]. However, we primarily focused on classification within species in our literature review, which bears closest resemblance to our problem.

2.1. Review

One of the earlier works in fine-grained classification was an attempt at identifying plant species by Belhumeur et. al. [6] This approach involved segmenting a leaf and then using shape to determine the species. Along similar lines, a paper by Farrell et. al attempted to identify a birds species by finding keypoints along the beak, eyes, wings, feet, and tail, and building features around them. [8]

More relevantly, however, a 2012 paper by Liu et. al attempted dog breed identification using a similar approach. [2] They first use an SVM regressor using greyscale SIFT descriptors as features to isolate the face of the dog. To handler rotation and scale, the window is also rotated and scaled; by using non-maximum suppression and picking the detection with the highest score, they isolate a single best window.

The primary focus of the paper is to find the facial keypoints of the dog. Liu et. al leverages a part localization algorithm, in which a sliding window SVM detector using

SIFT greyscale descriptors is used over each eye and nose. After the eyes and nose have been detected, greyscale SIFT descriptors around the keypoints are used as features by an SVM classifier. With this approach, Liu et. al is able to classify their test dataset with an accuracy of about 90

2.2. Comparison

Liu et. als results are certainly very impressive. However, significant effort is put into identifying the dogs face and then its keypoints. In fact, Liu et. al identify the failure of dog face detection as the primary bottleneck in their pipeline.

In our work, we attempt to use a convolutional neural network to assist with keypoint detection in dogs, namely identifying eyes, nose, and ears. CNNs have seen success in identifying facial keypoints in humans, and we hope to apply this technique to dogs as well. [4] By doing so, we eliminate dog face detection as a step in the process, and replace Liu et. als part localization process.

For classification, We utilize multinomial logistic regression and nearest neighbor models for classification, neither of which are used by Liu et. al. With our own novel approach, we hope to match the success of Liu et. al.

3. Technical Solution

3.1. Summary

To solve this fine-grained classification problem, we developed the analysis pipeline seen in figure 1. First, we trained a convolutional neural network on images of dogs and their annotated facial keypoints (right eye, left eye, nose, right ear tip, right ear base, head top, left ear base, and left ear tip). We used this network to then predict keypoints on an unseen test set of dog images. These predicted keypoints were then fed into a feature extraction system that used these keypoints to create more meaningful features from the image, which could later be used to classify the image. The primary features extracted were grayscale SIFT descriptors centered around the each eye, the nose, and the center of the face (the average of these 3 points). These features were then used for a variety of classification algorithms, namely bag of words, K-nearest neighbors, logistic regression, and SVM classifiers. These algorithms classify the images as one of 133 possible dog breeds in our dataset and complete the analysis pipeline.

3.2. Details

3.2.1 Keypoint Detection

To best identify dog breeds, the first step of analysis is keypoint detection. Dog face keypoints are defined and annotated in the Columbia dataset as the right eye, left eye, nose, right ear tip, right ear base, head top, left ear base, and left ear tip. Thus, the goal of this part of the analysis pipeline is defined as follows: given an unseen image from the testing set, predict the keypoints of the dog face as close as possible to the ground truth points in terms of pixels. Convolutional neural networks have been shown to perform quite well on a variety of image detection and classification tasks, including human face detection, but previous literature on dog face detection had not used neural networks; hence, we decided to tackle a novel approach for dog face keypoint detection. To solve this problem, we trained a fully connected convolutional neural network on 4,776 training images, which was used to predict the keypoints of 3,575 testing images. Before training, the images were all scaled to be 128x128 pixels; furthermore, the pixel intensity values were scaled to be in the range [0,1]. The ground truth keypoints were also scaled accordingly. The neural network was constructed using the nolearn lasagne API. [10]

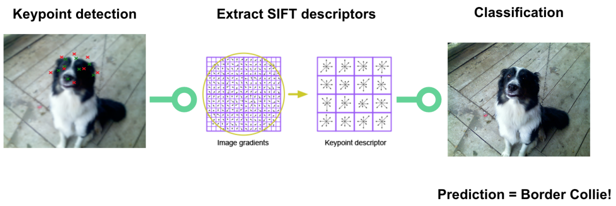


Figure 1. An image representation of our analysis pipeline

It performed regression using a mean squared error loss function defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y_i} - Y_i)^2$$

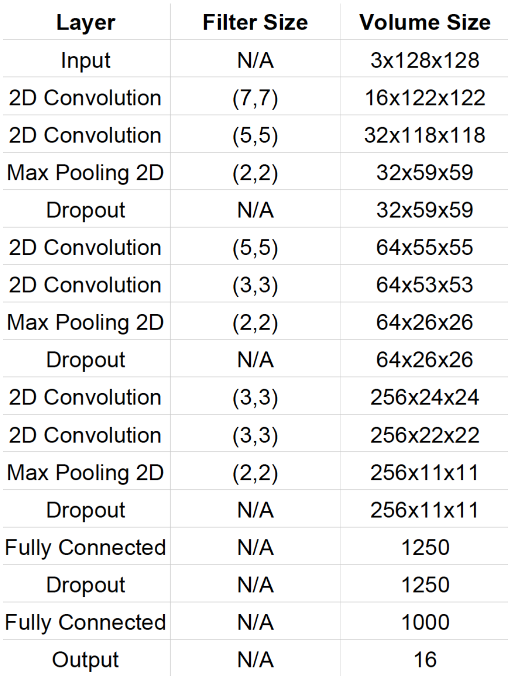
The network was trained using batches of 180 images for 4,000 epochs. In each batch, half of the images and their corresponding keypoints were flipped. This augmentation created a larger training set, which helped prevent the network from overfitting the data given. The architecture of the network went through several iterations, and the final construction is shown in figure 2.

3.2.2 Feature Extraction

Once we had detected the facial keypoints, we used them to extract meaningful features about the image, which we would later use in classification.

SIFT Descriptors

One feature we used was grayscale SIFT descriptors centered at important keypoints. SIFT descriptors do a good job representing localized regions of an image in a way that allows it to later be compared with localized parts of other images (e.g. we can compare one dogs eye to another and see if they seem similar). We centered our SIFT descriptors



Layer	Filter Size	Volume Size
Input	N/A	3x128x128
2D Convolution	(7,7)	16x122x122
2D Convolution	(5,5)	32x118x118
Max Pooling 2D	(2,2)	32x59x59
Dropout	N/A	32x59x59
2D Convolution	(5,5)	64x55x55
2D Convolution	(3,3)	64x53x53
Max Pooling 2D	(2,2)	64x26x26
Dropout	N/A	64x26x26
2D Convolution	(3,3)	256x24x24
2D Convolution	(3,3)	256x22x22
Max Pooling 2D	(2,2)	256x11x11
Dropout	N/A	256x11x11
Fully Connected	N/A	1250
Dropout	N/A	1250
Fully Connected	N/A	1000
Output	N/A	16

Figure 2. The architecture of our convolutional neural network

at the left and right eyes, the nose, and the center of the face (calculated as the center of these three points). We rotated our SIFT descriptors to match the rotation of the dogs face (calculated as the rotation away from horizontal for the line connecting the two eyes). We also sized our descriptors to be half the distance between the two eyes. We used Pythons cv2 library from OpenCV to calculate these SIFT descriptors.

RGB Histogram

Image color histograms are frequently used in image search engines. Given how dog coloring can vary, we believed it was worthwhile to explore using a color histogram as features for our classifier. To do so, we assumed an RGB color space, with pixel intensity values ranging from 0 to 255. As such, we had 256 bins (one for each pixel value), and three different histograms (one for each RGB channel). We limited the color histogram to the face of the dog to ensure that the background would not interfere. In total, this produced 768 features (256 bins × 3 channels).

Color Centers Histogram

The next color histogram we tried implementing used k-means to calculate the 32 color centers of all pixels over all training images (see figure 4). The feature we then extracted for each image was a histogram of pixel values where each bin corresponded to a color center and contained all pixels that were closest to that color center.

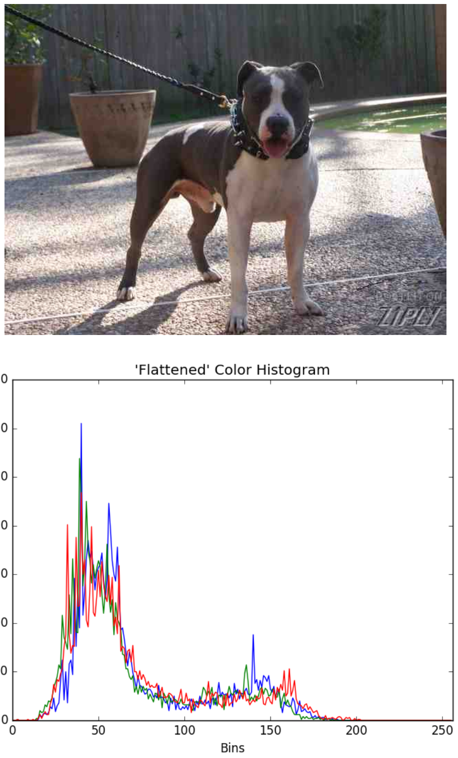


Figure 3. An American Staffordshire Terrier from our dataset and its corresponding color histogram (divided into RGB channels).

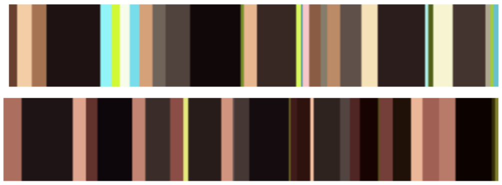


Figure 4. Color centers calculated using kmeans over the entire image (top) and just the dog's face (bottom) on the training set. The size of the bar represents how many pixels in the training set belong to that color center.

We realized we were getting a lot of noise from background colors, which should have no effect in determining a dogs breed, so we later refined this algorithm to consider only the pixels within the bounding box of the dogs face (calculated using the facial keypoints). We re-calculated the 32 color centers over only the pixels contained in the face (see figure 4), and re-calculated our histograms to also cover only those pixels within the bounding box of the dogs face.

3.2.3 Classification

Bag of Words Model

We decided to use a bag of words model for our base-line because it is a simple model that is frequently used for object classification. The bag of words model ignores our detected facial keypoints and instead just finds a visual vocabulary based on the cluster centers of SIFT descriptors obtained from the training set of images. Because bag of words models perform better classification for objects with very high inter-class variability, we did not expect it to work very well for our fine-grained classification problem.

For our bag of words model, we first used OpenCVs FeatureDetector and DescriptorExtractor to extract SIFT descriptors. We then used the K-means algorithm, from Pythons scipy library, to extract a visual vocabulary. We used this visual vocabulary to create feature histograms, preprocessed and scaled these histograms to fit a gaussian distribution, then used them to train an sklearn LinearSVC model.

SVM

The first real classification model we tried was an SVM. SVMs are commonly used machine learning models that perform well at multi-class classification. To date, few other supervised learning algorithms have outperformed SVMs, which is why we thought this would be a good place to start.

We tried a few different types of SVMs in order to determine which worked best for our use case. The first we used was the same LinearSVC model from sklearn that we used for our bag of words model. We then tried a normal SVM with a linear kernel and a OneVsRestClassifier, all using the same sklearn library for consistency. All SVMs used the grayscale SIFT features centered at facial keypoints and the color centers histogram calculated around the dogs face. In our experimental results section, you can see that the normal SVM with linear kernel outperformed the other two SVMs while keeping the features consistent.

Logistic Regression

As we learned in class, linear classifiers are a commonly used discriminative classifier for categorizing images. Unlike a standard linear regression problem, we have a finite number of discrete values (the various possible dog breeds) that our predicted value y can take on. To make our prediction, we would use the following hypothesis function:

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

Note, however, that the standard application of the sigmoid function for logistic regression is used for binary classification (where y can take on only two discrete values). As we have 133 possible breeds, we extrapolate the above logic to handle multiple classes, also known as multinomial logistic regression.

Instead of having a single logistic regression equation, we have $J - 1 = 132$ equations, where J represents the total number of breeds (we only need $J - 1$ since one of the breeds will serve as a reference). Thus, the log probability of dog i being breed j is:

$$\log P(i = j) = \alpha_j + x_i' \beta_j$$

where a_j is a constant, x_i is dog i's feature vector, and β_j is a vector of regression coefficients of breed j, for $j = 1, 2, ..., J - 1$.

Analogous to the standard logistic regression model, our model's probability distribution of response is multinomial instead of binomial, and we use standard Maximum Likelihood Estimation to predict the most likely breed. While such applications of logistic regression have been used before in fine-grained classification, our literature review showed no prior use in dog breed classification specifically. To implement, we used scikit-learn's $linear\_model$ module in Python.

K-Nearest Neighbors

Finally, we also attempted a nonlinear discriminative classifier using a K-nearest neighbor classifier. A dog is classified by a majority vote of its neighbors, with the dog being assigned to the breed most common among its k nearest neighbors. As noted in lecture, this method depends on training set being large enough to generate enough meaningful votes and does not always produce the optimal results. As such, we anticipated one of the linear classifiers performing better.

4. Experiments

Dataset

For our experiments, we used the Columbia Dogs Dataset as it provided the most robust data available online. The dataset contains 8,350 images of 133 different dog breeds some of which are featured in figure 5. Each image had a corresponding text file that annotated both bounding boxes and keypoints for each dog face. The facial keypoints annotated were the right eye, left eye, nose, right ear tip, right ear base, head top, left ear base, and left ear tip, which we used to train our convolutional neural network for keypoint detection.

Keypoint Detection

Qualitatively, the results of the keypoint prediction can be seen in the images in figure 6 where the red crosses are the predicted points and the green crosses are the ground truth points. Quantitatively, we evaluated keypoint detection based on the average distance between the ground truth keypoint and its predicted counterpart. On average, the neural network predicted the keypoint to be 4.62 pixels from the ground truth point.

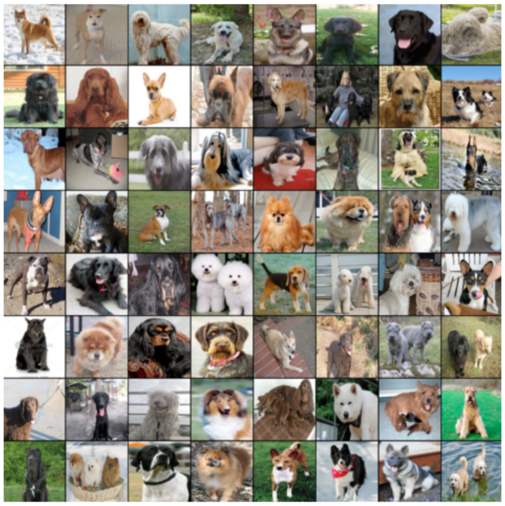


Figure 5. Example images from the Columbia University Dogs dataset.

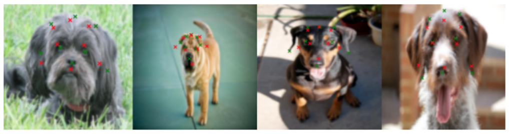


Figure 6. Qualitative results of our keypoint detection CNN.

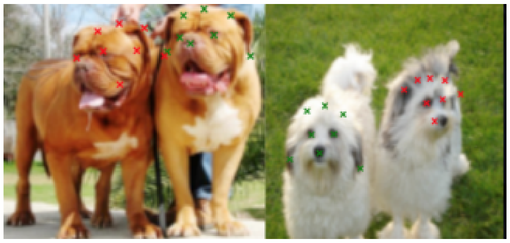


Figure 7. Example images where our convolutional neural network identified the ”wrong” dog in the image.

We noticed our convolutional neural network had trouble when more than one dog was featured in an image (it would sometimes try to identify the wrong dog, see figure 7). Additionally, the convolutional neural network sometimes had issues identifying the tips of the ears for some dog breeds (see figure 8). We figured this was because ear position is extremely variant among different dog species (can be perked up, or almost blend in with the rest of the fur on the head). As a result, we decided not to use the ears as SIFT keypoints in our classifiers, so as not to add noisy features to the feature set.

Classification

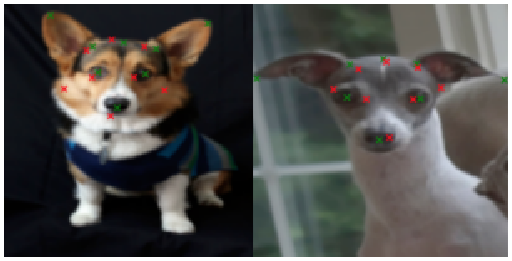


Figure 8. Examples of dog breeds where the convolutional neural network performed poorly at detecting the ear tips.

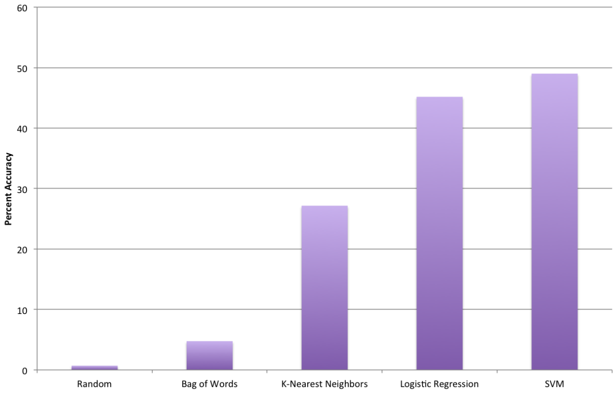


Figure 9. Comparison of the performance of different dog breed classifiers.

We ran each of our classifiers using our SIFT descriptor feature set and compared the accuracy of each model (see figure 9). While all models out-perform our control (randomly selecting from the 133 different breeds), the SVM and logistic regression models clearly produce the best results, obtaining 49% and 45% accuracy respectively. The K-nearest neighbors model performed at about half this accuracy while our bag of words model maintained only 5% accuracy results. We were not surprised by the poor performance of the bag of words model because the features for this model did not utilize facial keypoints like the other models did. Nearest neighbors is known to often produce suboptimal solutions, as it depends on the training set being large enough to generate meaningful votes. Note that multinomial logisti regression is not typically used in fine-grained classification, but we believe its success merits further research and consideration.

Looking more closely at the different types of SVMs we implemented, we found, as stated previously, that a normal SVM with linear kernel performed better than either a LinearSVC or a One-vs-rest SVM (see figure 10).

We were also interested in comparing how well our different features did relative to one another. Comparing different feature sets across our best performing model (the SVM with linear kernel), we found that our RGB histogram actually added noise to the feature set, while the color centers histogram (centered on dog faces) added useful infor-

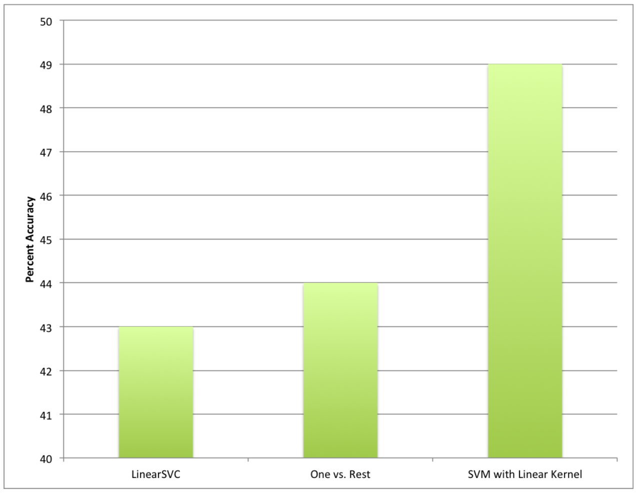


Figure 10. Comparison of the performance of different SVMs.

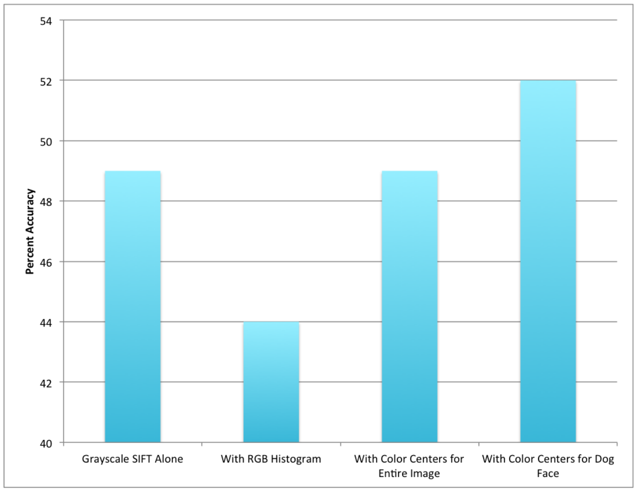


Figure 11. Comparison of the performance of different feature sets.

mation, albeit only by a few percentage points of accuracy increase (see figure 11).

We imagine that the color histogram wasn't as helpful as we thought it would be because of the high intra-class variation of colors within a given breed. For example, Labrador retrievers can be yellow, black, or brown, which would confuse our model when including very different color center histograms (see figure 12) Likewise, our RGB color histogram faced similar shortcomings, and could have been affected by inconsistencies in lighting across photos in our dataset.

The last metric we wanted to note was our Top 5 and Top 10 accuracy percentages (how often the correct breed was identified in our top 5 or top 10 guesses, see figure 13). We noted that the literature uses these metrics to assess the performance of their models because multi-class classification is difficult, and usually a human can do the last step of comparing images to identify the actual match (the difficult work is coming up with these names out of 133 different breeds, which is what our model accomplishes). Our Top 10 accuracy is 90%, which matches the accuracy achieved by Liu et. al.

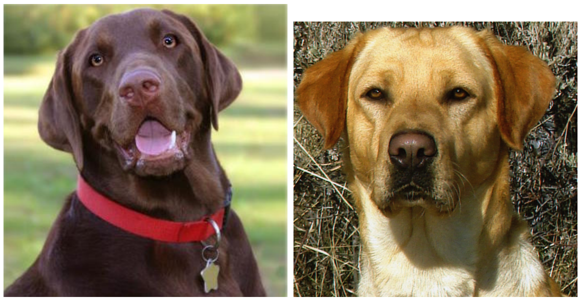


Figure 12. Brown and yellow Labrador retrievers, which should be classified as the same breed and are confounding our model when we add color-dependent features.

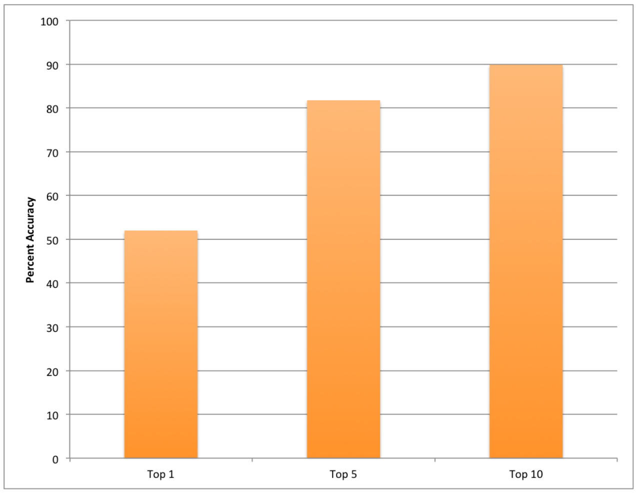


Figure 13. Top 5 and Top 10 accuracies of our best classifier model.

5. Conclusion

Overall, we consider our results to be a success given the high number of breeds in this fine-grained classification problem. We are able to effectively predict the correct breed over 50% of the time in one guess, a result that very few humans could match given the high variability both between and within the 133 different breeds contained in the dataset.

5.1. Contributions

We see two major contributions to the literature in this project. First, this is the first time deep learning and convolutional neural networks have been used for dog face keypoint detection according to the literature. Our keypoint predictions were 4.62 pixels from the ground truth points on average. The accuracy of our keypoint detection allowed us to succeed with our classification algorithms and is promising for future work in the area. Our vast experimentation with classification algorithms also provides novel contributions to the literature. The use of color histograms in feature extraction has never been done before; however, we found them to be unsuccessful due to the variety in color for individuals of the same breed. We also experimented with a variety of linear classifiers such as logistic regression and K-nearest neighbors for predicting dog breeds. These contributions add a variety of techniques to the literature for dog breed identification some which should be explored further as will be discussed next.

5.2. Future Work

Future work should further explore the potential of convolutional neural networks in dog breed prediction. Given the success of our keypoint detection network, this is a promising technique for future projects. That said, neural networks take an enormous time to train and we were unable to perform many iterations on our technique due to time constraints. We recommend further exploration into neural networks for keypoint detection, specifically by training networks with a different architecture and batch iterator to see what approaches might have greater success. Also, given our success with neural networks and keypoint detection, we recommend implementing a neural network for breed classification as well since this has not been performed in the literature. We were unable to experiment with this approach due to the time constraints of neural networks but believe that they would match if not improve upon our classification results. Ultimately, neural networks are time consuming to train and iterate upon, which should be kept in consideration for future efforts; still, neural networks are formidable classifiers that will increase prediction accuracy over more traditional techniques.

See our code here:
`https://www.dropbox.com/sh/nfg3xaqsvt099vb/AABhxuO5fecOBLaa_RFBT_0ea?dl=0`

References

[1] A. Angelova and S. Zhu. Efficient object detection and segmentation for fine-grained recognition. *2013 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[2] J. L. et al. Dog breed classification using part localization. *Computer Vision: ECCV 2012*, pages 172–185, 2012.

[3] N. Z. et al. Deformable part descriptors for fine-grained recognition and attribute prediction. *2013 IEEE International Conference on Computer Vision*, 2013.

[4] N. Z. et al. Part-based r-cnns for fine-grained category detection. *Computer Vision: ECCV 2014*, pages 834–849, 2014.

[5] O. M. P. et al. Cats and dogs. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[6] P. N. B. et al. Searching the world's herbaria: A system for visual identification of plant species. 2008.

[7] P. N. B. et al. Localizing parts of faces using a consensus of exemplars, 2013.

[8] R. F. et al. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. *Pattern Recognition: 36th German Conference, GCPR 2014*, 2014.

[9] E. Gavves. Fine-grained categorization by alignments, 2013.

[10] D. Nouri. Using convolutional neural nets to detect facial keypoints tutorial, 2014.