EECS 442: Fine-grained Dog Breed Classification

Lina Zhang, Wen He and Zihan Li

Abstract— **In this report, a system for automatically identifying dog breeds via images is explained, implemented, and evaluated. The system consists of three major stages: facial key-point localization, facial normalization, and breed classification. We used convolutional neural networks (CNN) to predict facial key-point locations. After that we extracted SIFT features and color histogram based on predicted facial key-point locations. Further, we used extracted new features as input of linear SVM to classify dog breed. All three stages are crucial to the performance and understanding of the system as a whole. Results on the breed identification task are compared to those of Liu et al, 2012 [1].**

I. INTRODUCTION

In most cases, image classification follows a common pipeline in which a set of features are extracted from an image and fed to a classifier. Based on related works on fine-grained dog breed classification, we know that localized feature key-points will provide better information for classifier than those using global features [1]. Recently there is a growing trend in classifying pictures directly with CNN. We wanted to examine whether a pure CNN classifier outperforms localized features method.

For our purpose, we used Columbia Dogs as our dataset[1], which contains 8,351 image dataset with not only class labels for 133 dog breeds but also 66,808 part labels [1]. Using the dataset, we were able to train a CNN to predict key facial point given an image and extract SIFT features around the points.

In this project, we applied what we learned in class and implemented: a CNN to find facial key points, a feature extractor to get SIFT features and color histogram, a SVM classifier to classify dog breed based on SIFT features and color histogram, and another CNN to directly classify dog breed. With our test, we argue that with the same CNN structure and training time, localized features classifier out performs pure CNN classifiers [2].

II. RELATED WORK

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 [1]. 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 key-points. 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 gray-scale SIFT descriptors. The key-point 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.

In contrast, work by LaRow W. [2] used CNN instead of sliding window SVM for facial key-point detection. Their accuracy was generally lower than the accuracy of Lius [1] breed identification algorithm. However, we chose their structure to gain experience in building and training CNN to classify breeds and detect key points.

III. METHODS

A. Baseline: CNN

We use convolutional neural network for classification as baseline. The input images were resized to 128x128. The pixel intensities of input images were scaled to $[0,1]$ range. We use categorical cross entropy as loss function. The loss function is formulated as

$$H(p,q) = -\sum_{x} p(x)\log(q(x))$$

where p is true distribution and q id coding distribution. The network was trained using batches of 180 images for 800 epoches. The architecture of the network is shown in Table I. The network was constructed using the Keras API.

[1] The dataset of Columbia Dogs can be downloaded here.

[2] The GitHub repository for the project is here.

Layer	Filter Size	Volume Size
Input	N/A	128x128x3
Convolution	(7,7)	122x122x16
Convolution	(5,5)	118x118x32
Max Pooling	(2,2)	59x59x32
Dropout	N/A	59x59x32
Convolution	(5,5)	55x55x64
Convolution	(3,3)	53x53x64
Max Pooling	(2,2)	26x26x64
Dropout	N/A	26x26x64
Convolution	(3,3)	24x24x256
Convolution	(3,3)	22x22x256
Max Pooling	(2,2)	11x11x256
Dropout	N/A	11x11x256
Fully Connected	N/A	1250
Dropout	N/A	1250
Fully Connected	N/A	1000
Output	N/A	133

TABLE I
ARCHITECTURE OF CNN FOR CLASSIFICATION.

B. Key point Detection: CNN

Instead of using CNN for classification directly, we use it to locate dog face key points before applying descriptors at these key points for classification task. Dog face key points locations are given in Columbia dataset as right eye, left eye, nose, right ear tip, right ear base, head top, left ear base and left ear tip (8 points per image). We trained a CNN on the training set of 4,776 images to predict the positions of the 8 key points. The input images were resized to 128x128 and the ground truth key points coordinates were recalculated accordingly. The pixel values of input images were scaled to $[0,1]$ range. We use mean squared error as loss function, since our goal is minimizing distance between predicted points and ground truth points. The loss function is formulated 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 800 epoches. The architecture of the network is the same as in Table I except that the output is a 16-dimensional vector. The network was constructed using the Keras API.

C. Features

1) SIFT: SIFT descriptor does a good job in representing local features and is scale and rotation invariant, so we use it as features for classification. We use the positions of the left and right eyes and the nose from the located 8 key points to calculate SIFT descriptors. We added another key point at the center of dog face, which is the average of left and right eyes and nose. The SIFT descriptors are rotated to align the line connecting two eyes. The diameter of SIFT descriptors is half the distance between two eyes. Python cv2 library was used to calculate SIFT descriptors. This produced 512 features (128 x 4 key points).

2) Color Histogram: In addition to SIFT descriptors, we also use color histogram as features, because we believe color plays an important role in classifying dog breed. The color histogram is only calculated from dog's face. The mask is a rectangle with the average of eyes and nose as center and the distance between eyes as half width. We calculated a histogram of 32 bins for each RGB channel. The color histogram vector is normalized since masks are of different sizes for different images. Python cv2 library was used to calculate color histogram. This produced 96 features (32 bins x 3 channels). The color features along with SIFT features are concatenated to produce a 608-dimensional feature vector.

D. Classification: SVM

SVM performs well at multi-class classification. SVM is mentioned in LaRow et al, 2012 [2] to have outperformed other classifiers such as Logistic Regression and K-Nearest Neighbors in similar task, so we adopted SVM in classification. The input of SVM classifier is the 608-dimensional feature vector which is calculated in previous section. We tried two settings of SVM: one SVC from sklearn library with linear kernel and the other with RBF kernel. The one with linear kernel outperformed the other in a significant amount. It may partly because we did not figure out the right hyperparameters for the RBF kenel SVC.

IV. RESULTS

We are dividing the results of our project into three parts. In the first part, we focus on examining the localization of face parts through constructing visualization. In the second part, we test the SVM classification based on the detected face features. Finally, we construct the full test that runs through CNN, SIFT ans SVM as described before. The accuracy is reported based on the original classification in the dataset.

A. Localization of Face Parts

To get an intuitive evaluation of the our CNN model performance on detecting face parts on different dogs, we plotted face parts onto the dog image data including right eye, left eye, nose, right ear tip, right ear base (inner base), head top, left ear base (inner base) and left ear tip. As is shown in following figures, the performance of CNN model on images with high quality dog faces is pretty good (Fig. 1), whereas the performance for those images with lots of noise (other dog-like subjects) or with dogs of strange poses (Fig. 2) is not as good as the former case.

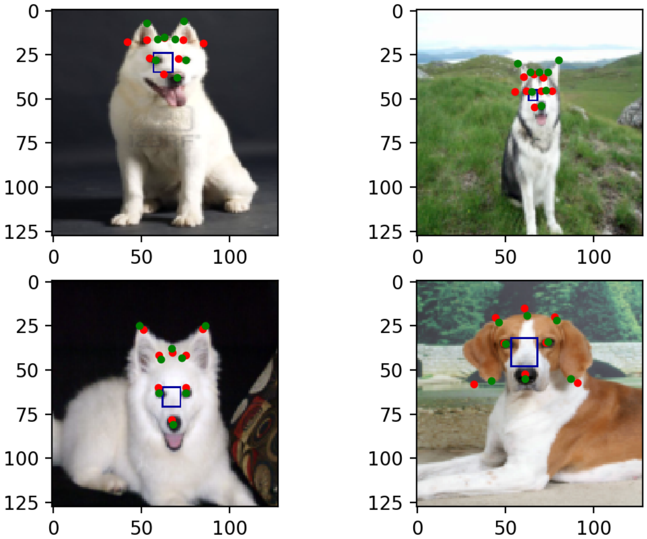

Fig. 1. Four samples that perform good on CNN face feature detections. The green dots are the correct label of feature points and the red ones are detected by CNN model.

B. SVM classification Based on Detected Face Parts

1) Model and Hyperparameter selection: We selected our SVM model based on the ground truth of feature points provided in the dataset. We have tried different kernels of SVM including linear and RBF, as well as values of hyperparameters including the penalty parameter C and kernel coefficient γ of SVM model on the cross-validation dataset. The results shown on the cross-validation set indicates that the linear kernel is superior to and more stable than the RBF kernel among different values of hyperparameters. The accuracy of linear kernel is around 51% in the case when penalty parameter $C \in [0.1, 100]$, whereas the accuracy of RBF kernel varies with C and γ, with best accuracy around 48% when $C = 50, \gamma = 10^{-7}$. Figure 3 illustrates the trend between accuracy and values of hyperparameters.

2) Performance Visualization: Here we show a sample result for our dog breed classification, including six classes each with five examples. As can be observed from Fig. 4, the overall performance of classification is good. Some inconsistencies may be observed but it is due to the noises in the original dataset.

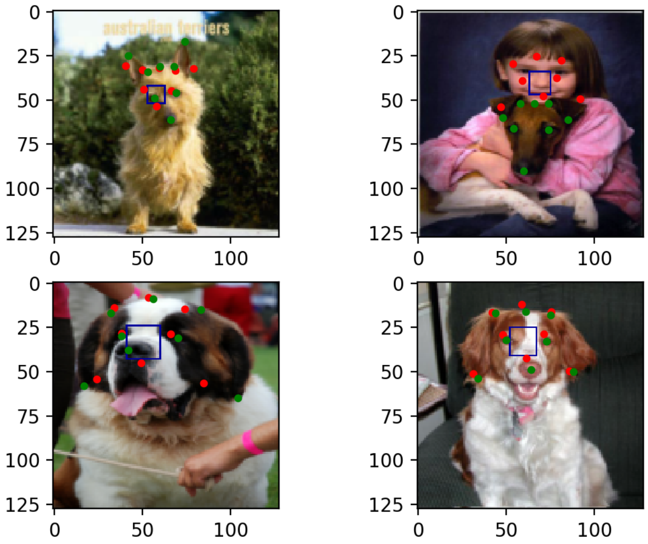

Fig. 2. Four samples that perform not so good on CNN face feature detections. Note that in the second image, the feature point detection is made falsely on the face of the girl.

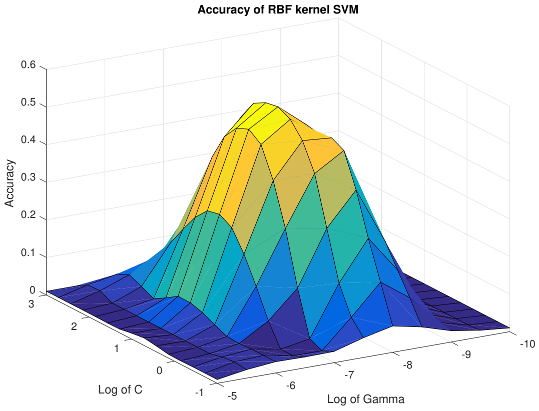

Fig. 3. The accuracy on the cross-validation data set with different values of hyperparameters (C and γ) assigned.

C. Performance on the Whole System

Our dog breed classification system reaches the accuracy of 37% on the test set. Note that this is expected to be lower than the accuracy provided in the previous section since it is evaluated on a separate test set. Our classification system performs pretty well on images with dogs of right pose and good lighting condition. For those images that contain dogs' owner or even more dogs, the system does not perform as well as the first case.

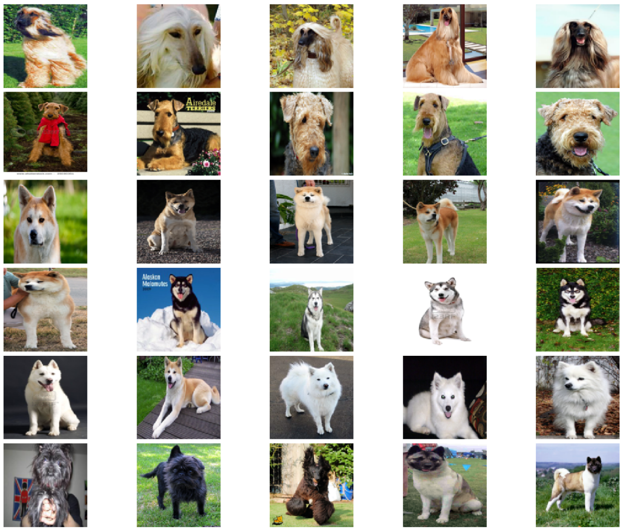

Fig. 4. Classification results from SVM model. Six classes of dog breeds with one row including five dog sample images are included here. Breeds from the first row to the last row are Afghan Hound, Airedale Terrier, Akita, Alaskan Malamute, American Eskimo Dog, and Belgian Tervuren

We have also tested the performance on system that only applies CNN to do dog breed classification. The accuracy is 23%, which is lower than the current system. It indicates that removing background noises is crucial in bringing up the classification accuracy. The local face parts means more to the dog breed classification. The accuracy comparison between methods stated above is visualized in Fig. 5.

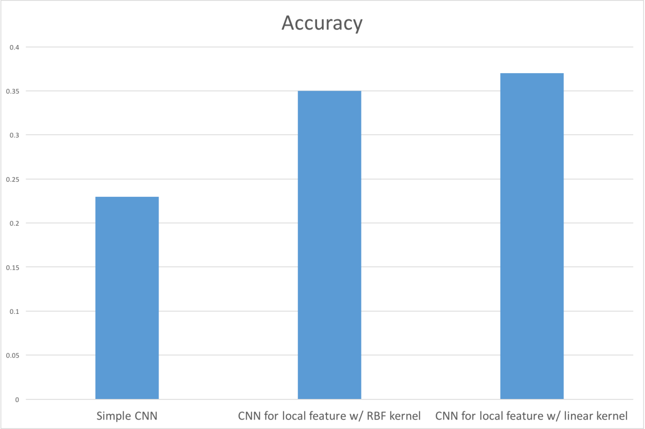

Fig. 5. Accuracy comparison between simple CNN structure and CNN for local feature detection with RBF kernel and linear kernel.

V. DISCUSSION & CONCLUSIONS

This dog breed classification system does not perform as well as the one mentioned in Liu's paper (67%), but on the same level as the one done by Bag of Words (BoW) model [1]. The result is acceptable taking the short time and simplified model into consideration. Comparing with Liu's paper, we did not use RBF kernel in SVM fitting because we could not find proper hyperparameter doing better than the linear kernel one. We made use of CNN method when finding feature face parts.

There are many places to be improved. The CNN structure should be further evaluated and improved. We should also seek to find potential ways that may perform even better than CNN. We can also try to find the localization feature among the face parts detected since different breeds of dogs have different face parts relative location. One other thing that we could improve is finding deeper significance in the underlying color histogram.

This project has great application significance since the same idea can be applied to classification of other subjects. It can also be combined with other larger-scale applications. One of the things that may hinder the system from growing to more areas is that it requires prior manual label. In the future, we want to explore more unsupervised learning techniques so that the manual labeling procedure can be omitted.

ACKNOWLEDGMENT

We want to thank Professor Jason Corso for his advice. We also received great help from the Graduate Student Instructor Siyuan Chen and Kyle Byungsu Min of the course Computer Vision (EECS 442).

REFERENCES

[1] Liu J., Kanazawa A., Jacobs D., Belhumeur P. *Dog Breed Classification Using Part Localization.*. In: Fitzgibbon A., Lazebnik S., Perona P., Sato Y., Schmid C. (eds) Computer Vision ECCV 2012. ECCV 2012. Lecture Notes in Computer Science, vol 7572. Springer, Berlin, Heidelberg.

[2] LaRow W., Mittl B., Singh V. *Dog Breed Identification.*.