Recognizing Cats and Dogs with Shape and Appearance based Models

Group Member: Chu Wang, Landu Jiang

Abstract

Recognizing cats and dogs from images is a challenging competition raised by Kaggle platform to crack a Pet-CAPTCHA raised by Asirra, which asks users to identify cat and dog photographs with large variation in size and noise. Humans can accomplish it quickly and accurately, however, to train an artificial intelligent system coupling this task is non-trivial. In this project, we implement shape based deformable part model (DefPM)[4], which is suitable for a large range of object class, as well as the famous appearance model Bag of Visual Words. We train the DefPM using face and body annotations to recognize cats and dogs in Asirra images. This model achieved a good accuracy of 90.5% with 10 over 70 ranking on Kaggle Leaderboard and outperforms the bag of words approach which is based on SIFT features whose best accuracy is 81.93%, in comparison.

Key Words: Pattern Recognition, CAPTCHA, Deformable Part Model, SIFT, Bag of Words

The Kaggle Leaderboard:

Dashboard

Leaderboard - Dogs vs. Cats

This leaderboard is calculated on approximately 30% of the test data.
The final results will be based on the other 70%, so the final standings may be different.

See someone using multiple accounts?
Let us know.

#	Δ1w	Team Name * in the money	Score	Entries	Last Submission UTC (Best – Last Submission)
1	-	DaggerFS *	0.97040	4	Mon, 18 Nov 2013 17:54:09 (-3d)
2	-	Charlie *	0.96987	5	Sat, 16 Nov 2013 17:30:20 (-30.6d)
3	-	Jeff	0.96773	2	Thu, 26 Sep 2013 13:54:31
4	-	wqren	0.96667	4	Wed, 23 Oct 2013 02:12:19 (-1.1h)
5	↑1	Kyle Kastner	0.96667	10	Mon, 25 Nov 2013 15:17:11 (-4.1d)
6	↓1	Daniel Nouri	0.96587	5	Sat, 23 Nov 2013 20:43:31 (-5d)
7	↑1	hungry red panda	0.96427	2	Wed, 27 Nov 2013 09:17:39
8	↓1	naxeji	0.94933	10	Thu, 24 Oct 2013 19:00:57 (-21h)
9	-	Poly	0.92000	4	Sun, 17 Nov 2013 09:19:38 (-4d)
10	new	**FATE**	**0.90560**	**2**	**Tue, 26 Nov 2013 20:09:50**

I. Introduction

With the increased usage of Internet in every sector of human society, web services are becoming increasingly vulnerable to attacks. Current technologies are often protecting them with a challenge that's supposed to be easy for people to solve, but difficult for computers. Such a challenge is often called a CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart), and comes into large variations in question content, such as words, arithmetic, image or even riddles.

Recently, Kaggle hosted a competition to challenge Asirra's Image CAPTCHA, which is a dataset containing 40000 cat and dog images, and called for the state-of-the-art accuracy of recognition. In this project, we accepted this competition and implemented the shape based deformable part model [4] as well as the appearance based bag of visual words model [9] to construct the raw pixel image features. Then we used SVMs as well as Neural Nets to classify the constructed features and get the final classification result.

Our concentration lies in the shape based Deformable Part Model within which we run extensive training and evaluations while treat the appearance models as a good contrast method with few analysis since the relative literature is already a lot out there.

In the next paragraphs, we will first introduce the dataset we utilized in this project and give explanation of the two feature construction models we applied.

i. Dataset Present

PASCAL VOC 2009

Dataset provided by PASCAL VOC challenge 2009, with multiple object classes and corresponding body annotations, within which cat and dog are included. Besides, negative image containing no target object is also included in this dataset, which plays an important role in our further process of recognition. Sample image is displayed in Figure 1.1.

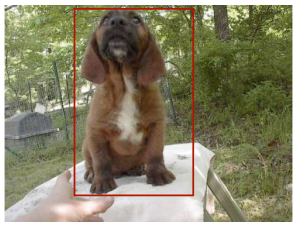
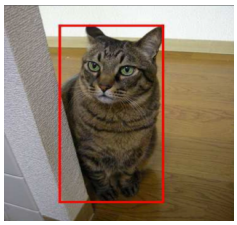
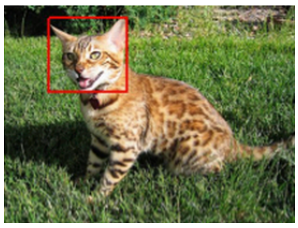
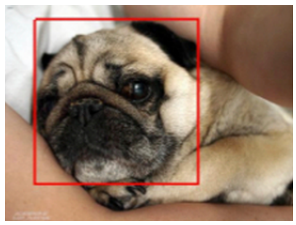





Figure 1.1 PASCAL VOC

Figure 1.2 Oxford IIIT Pet Dataset

Oxford-IIIT Pet dataset [3]

The Oxford-IIIT Pet dataset is a collection of 7349 images of cats and dogs. Every image is provided a head annotation in contrast to the PASCAL VOC dataset. It is designed for a much tougher problem, which is the breed classification. There are 37 different breeds, of which 25 are dogs and 12 are cats, each breed contains about 200 images. Sample image is displayed in Figure 1.2.

Asirra CAPTCHA Dataset [2]

This is the target dataset in this project. It contains raw image of cats and dogs coming in large variations in image size, content (body or half or head), noisy background. It is supported with 25000 training image and 12500 test image, and displayed in Figure 1.3.

Figure 1.3 Asirra CAPTCHA Dataset

ii. The Model for Feature Construction

In [5][6], Parkhi addressed the problem of classifying cats and dogs using appearance and shape based models. In this project, we followed their idea and implemented shape model as well as appearance model. We use the feature constructed by the two models of difference category as our classifier input and did extensive classifier training to suit the new feature space.

Considering the shape model, we referred to the Deformable Part Model [4], which is based on HOG filters [7] and proposed by P. F. Felzenszwalb in 2010. We applied the deformable part model on both head part of a pet and the corresponding

body part to train two different DefPMs. We process the target dataset of Asirra CAPTCHA with the two DefPM and classify the images based on new processed feature set. Finally we give comprehensive comparison and evaluation over the body model and head model.

To represent pets' appearance, we use a bag-of-words [8] model capturing the material of the pets' textures (e.g. color) and the object discontinuities (e.g. edges). As same as the shape-based model, the head provides a cue on the color of cats and dogs, and image pixels far enough from the head are used to estimate the color of the background. Image features are quantized based on a vocabulary of 4000 visual words [9][11] learned by using k-means method and then computed densely across the image by extracting SIFT descriptors. Quantized SIFT features are then recorded in a spatial histogram for each tile with dimension equal to 4000 times the number of spatial bins. The final feature vector for the image is a concatenation of these histograms and was trained by a linear support vector machine (SVM) as well as a 1-layer neural network. The process of appearance-based model is introduced with full detail in Section 4.

The rest of the report is organized as follows. Section 2 and 3 give detail description of shape based DefPM and appearance based BoW methods, which briefly introduce their Characteristics and Challenges. Then we present the classification scheme and experiments results in Section 4. The evaluation part is provided in Section 5. Section 6 concludes the project and discussed future work of our research.

II. Shape Feature: Deformable Part Model

In this section, we will introduce the deformable part model [4] (DefPM), which is the shape based model we applied to classify the Asirra Dataset.

We first present our solution map of using DefPM to arrive the classification result of Asirra Dataset. Then we introduce the feature construction process with DefPM as well as display our well-trained shape model of head DefPM and body DefPM. Finally, we briefly demonstrate the training process of Deformable Part

Model based on latent SVM.

In this section, for purpose of illustration, the DefPM we referred to is head deformable part model without specifically clarification. Head models are more computational friendly and generating better performance, which we will indicate in further evaluation section. For more information on body DefPMs, please check with our submitted code and files where we included our DefPMs in Matlab files.

i. Solution Methodology

We first present our pattern recognition system framework. Our system has below workflow steps:

- We first use head annotated dataset which is Oxford IIIT Pet Dataset to train the head DefPM. For the body DefPM, we use the PASCAL VOC dataset who has body annotations for training purpose.
- We then achieve the trained DefPM.
- We use the trained DefPM to process the Asirra Dataset. Each image is processed twice with cat model and dog model respectively.
- We get head annotations (or body ones if using body DefPM) and the scores (cat score and dog score).
- We use the training set cat score and dog score together with training labels to train SVMs to classify image with the constructed shape scores.
- We use the trained SVM to classify test set scores and get the test set labels.

We demonstrate the above process in the below flow chart:

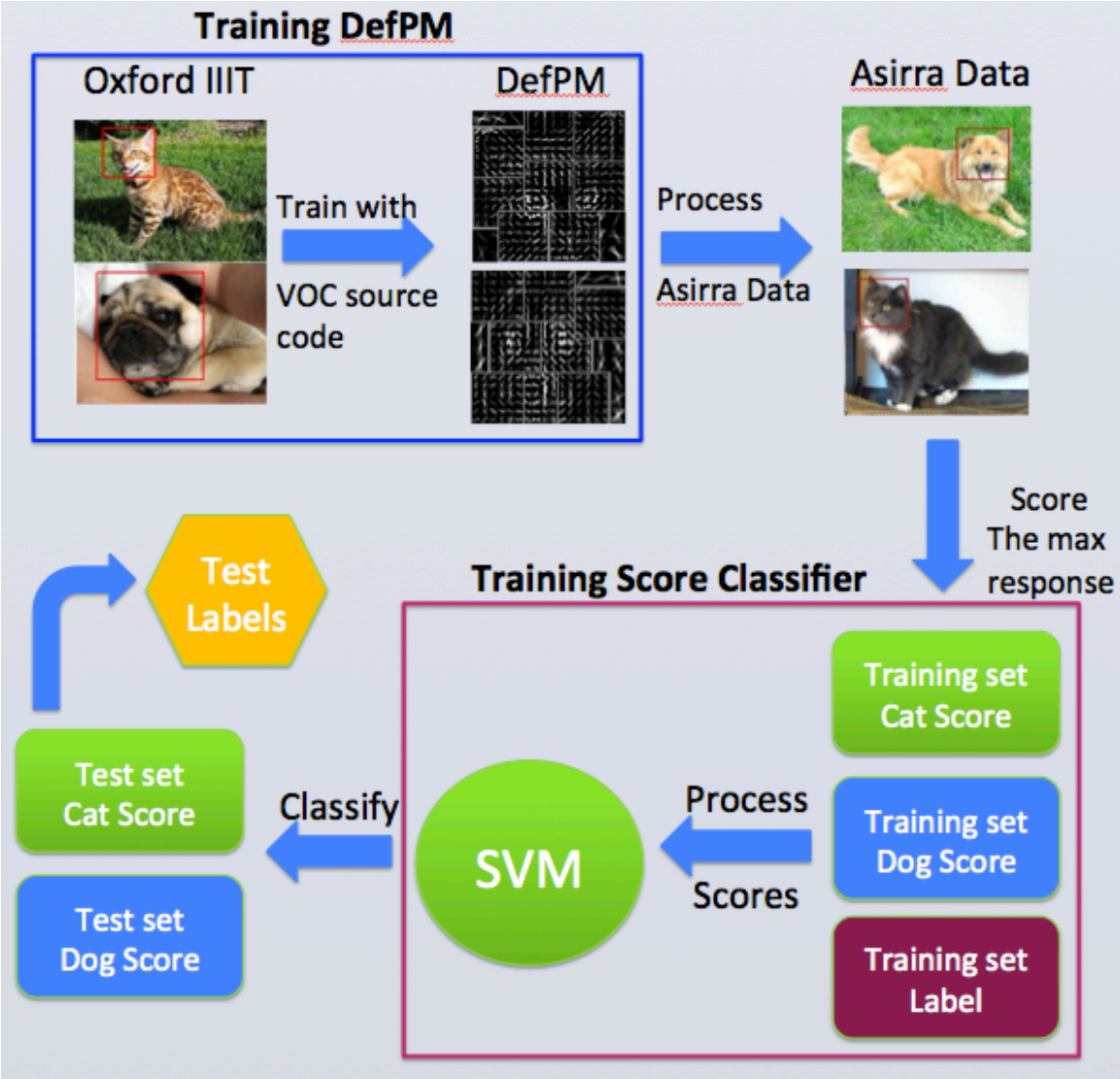
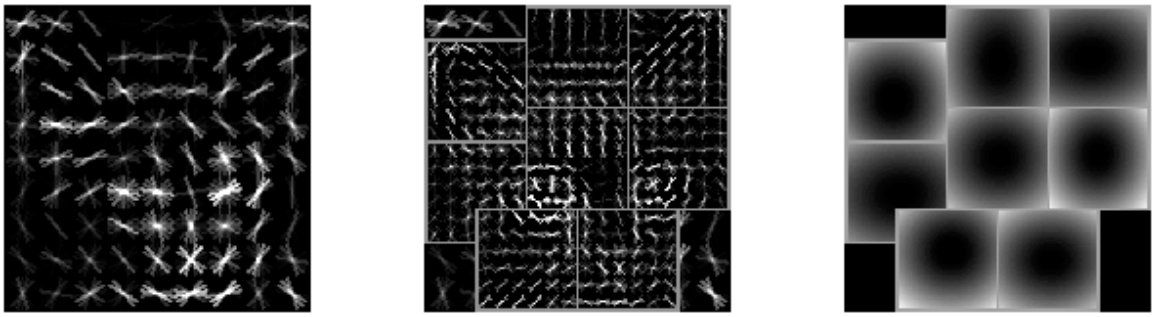


Figure 2.1 Solution Methodology for DefPM

ii. Score the Image: Feature Construction

The deformable part model is a set of HOG feature filters, among which there is one low resolution root filter and several high resolution part filters, and we display our trained deformable part model as below.



(a)root filter (b)part filter (c) part position

Figure 2.2 Cat Head Deformable Part Model

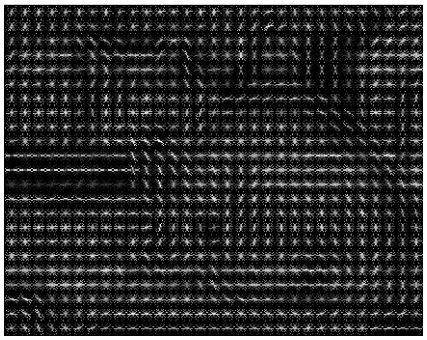
(a)root filter (b)part filter (c) part position

Figure 2.3 Dog Head Deformable Part Model

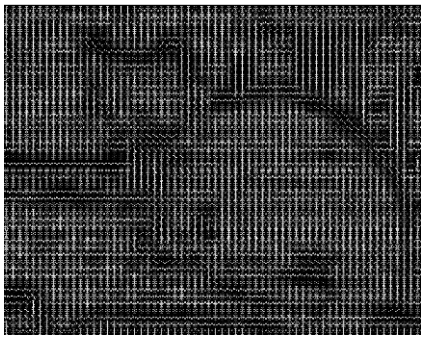
The scoring process of one image:

- A query image arrives.

- Convert the query image to HOG feature space, with 1x resolution and 2x resolution.

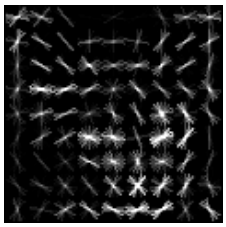


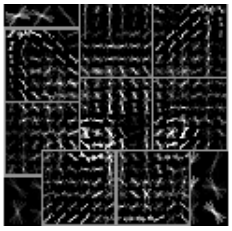
1x Resolution HOG



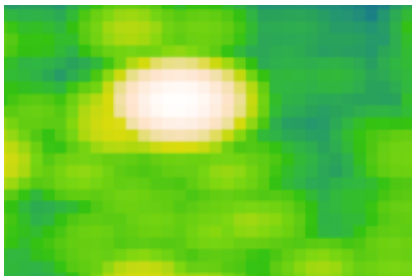
2x Resolution HOG

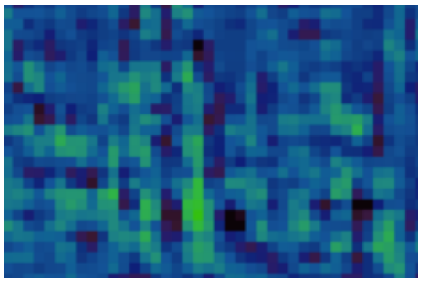
- Convolve the object root filter at the HOG feature map of original image and convolve the part filter with twice resolution.



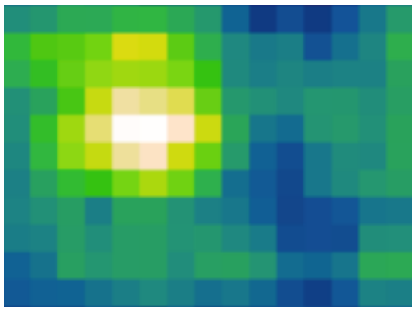


- Get the response map of two convolutions.

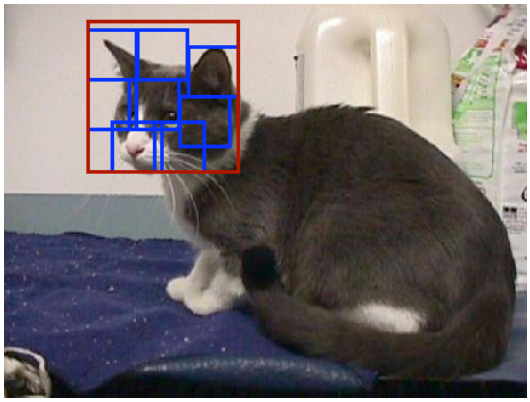




- Combine the response of two convolutions into one response map.



- Locate the instance area where the response is highest and formulate boundary box. Red for root filter, blue for part filters.



- Record the max response as the **cat score** of object.
- Repeat above process with dog head DefPM. Record **dog score**.

iii. Training the Deformable Part Model

We explain the process and basic theory in the deformable part model, to check with the full version, please refer to the reference [4]. We first define the related parameters and introduce the important training algorithm of latent SVMs. Finally we will give the pseudo code of implementation the latent SVMs.

Definitions of Parameters

- DefPM has parameter vector β : the raw vector reshaping of the 2-d HOG filter.
- The feature space x : the query image's HOG feature vector reshaped from 2-d version.
- Latent value: the bounding box locator descriptor $z \in Z(x)$
- The feature vector $\Phi(x,z)$: the feature space bounded by latent value.
- The score of model β over feature space x with latent value z :

$$f_{\beta}(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x,z)$$

We now face a problem of solving best model vector beta for all training images with latent value of boundary box in every training instance. This situation is well issued by latent SVM where we encounter 2 unknowns in training a SVM.

Latent SVM

In analogy to classical SVMs we train β from labeled examples $D = (< x_1, y_1 >, ..., < x_n, y_n >)$, where $y_i \in \{-1, 1\}$ by minimizing the objective function formulated by Lagrange Multipliers:

$$L_D(\beta) = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i))$$

The labeled examples are achieved from below dataset:

- Negative examples. The PASCAL VOC dataset provides negative images with no target object within them. Thus the negative examples are $N = \{J_1, J_2, ..., J_m\}$ which are labeled "-1".
- Positive examples. To train body DefPM, we use PASCAL VOC cat and dog images with body annotations. To train the head DefPM, we use Oxford IIIT Pet Dataset with head-annotated cat and dog images. In both circumstances, we define positive images as $P = \{(I_1, B_1), ..., (I_n, B_n)\}$, where B represents the annotation. They are all labeled with "1".

Training the Latent SVMs

We use the open source code available at [12]. To compile and use the source code, we made proper amendment to the compilers and parameter settings to make the code workable in our datasets. Details about our code and usage tutorial is

included in our code submission. Please refer to them for more support.

Training algorithm is displayed in below figure (Pseudo code):

0 while(t<Num_iterations)
1 for i=1 to n do
2 Solve the latent value which covers 50% of bounding box.
3 $z_p(i) = \arg\max_{z \in Z(x_i)} \beta \cdot \Phi(I_i, z)$
$st. \quad Box_\beta(z_i) \bigcap B_i \geq 0.5$
4 Add to
5 $F_p(i) = (x_i, z_p(i), y_i)$
6 end
7 for i=1 to m do
8 Solve the latent value
9 $z_n(i) = \arg\max_{z \in Z(x_i)} \beta \cdot \Phi(J_i, z)$
10 Add to
11 $F_n(i) = (x_i, z_n(i), y_i)$
12 end
13 Stochastic gradient
14 $\beta = gradient - descent(F_p \bigcup F_n)$
15 end

Figure 2.4 Pseudo code of Training DefPM

III. Appearance: Bag of Visual Words

In this section we overview the concept of a Bag of visual words vocabulary that enables efficient indexing for local image features.

i. Solution Methodology

Alike we did in Section 2, the basic process is summarized in the figure below:

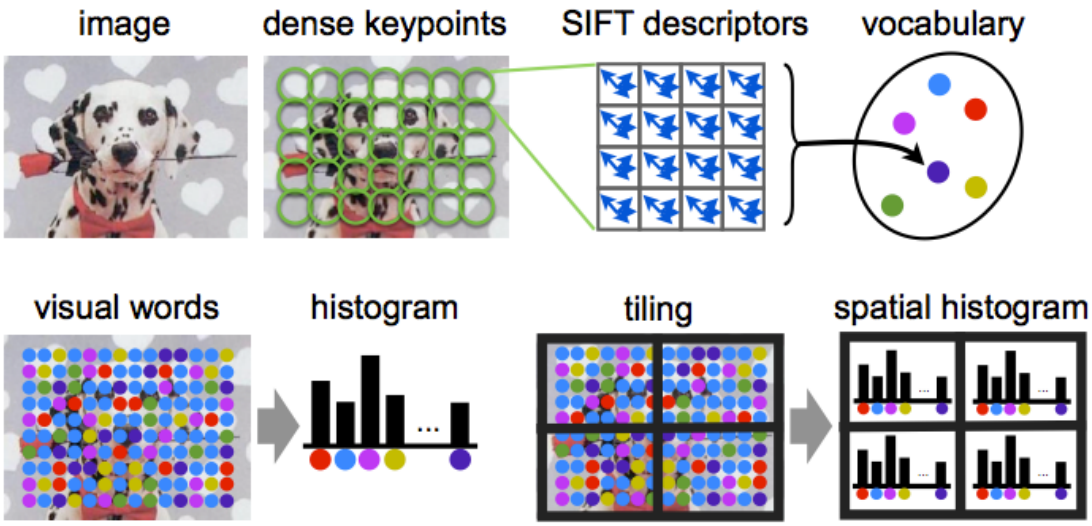


Figure 3.1 Example images of Head + Body Solution

We used the source code available [9][11], and made necessary amendment for our specialized objective.

ii. Creating a Visual Vocabulary

Text documents contain some distribution of words, and thus can be count as their frequencies (known as bag-of-words). To take the aid of text processing on visual search and obtain discrete "visual words", we must impose a quantization on the feature space of novel image descriptors (SIFT in our project). There are usually two standard steps to form a visual vocabulary: firstly, collecting a large sample of features from a representative image dataset (Asirra and Oxford-IIIT Pet datasets in this project, and (2) feature space quantizing, the simple k-means clustering is often used for the quantization. In this case, the size of the visual vocabulary is a user-supplied parameter defined as k. Once the vocabulary is established, the image's features can be translated into words by determining nearest distances between their neighbors. In general, patches assigned to the same visual word should have similar low-level appearance [10] (shown in Fig. 3.2).

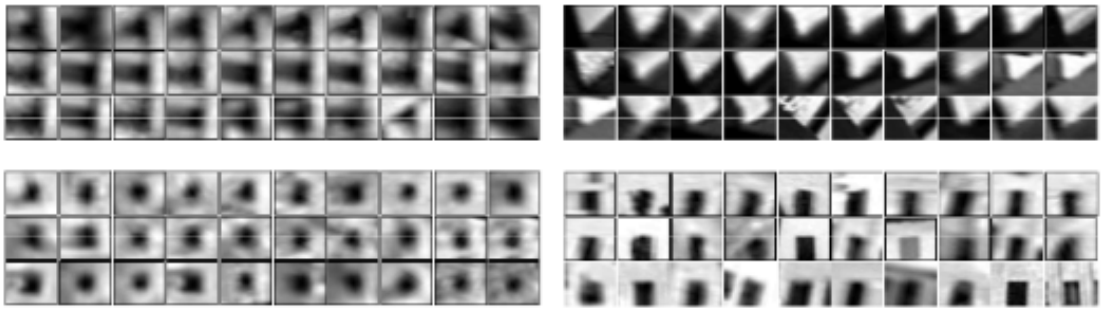


Figure 3.2 Four examples of visual words

iii. Creation of Histogram

The bag of words visual vocabulary enables a compact summarization across an image's words with an empirical distribution histogram in the visual vocabulary. It allows us to use many machine learning algorithms by translating a large set of high-dimensional local descriptors into a single sparse vector of fixed dimensionality. In our project, we firstly extract a number of visual descriptors from the specified images and clusters the descriptors into NUMWORDS visual words by using KMEANS. It then computes a KDTREE to index them. According to [9][11], the use of a KDTREE is optional, but speeds-up quantization significantly. Finally we build the distribution histogram using the visual vocabulary for final classification and return the image patches that the classifier thinks are most related to the class as shown in Fig. 3.3.

Figure3.3 BoW Classificaion Output Display

IV. Classification of Constructed Features

We present our training process of classifiers based on two categories of constructed features: shape features and appearance features. We separate them into two sub-sections.

i. Score classifiers for DefPM

In this section, we need to train classifiers to distinguish the constructed shape features which are the scores we achieved by DefPM. We use SVM for their effectiveness and displayable decision boundary.

Statement of Task

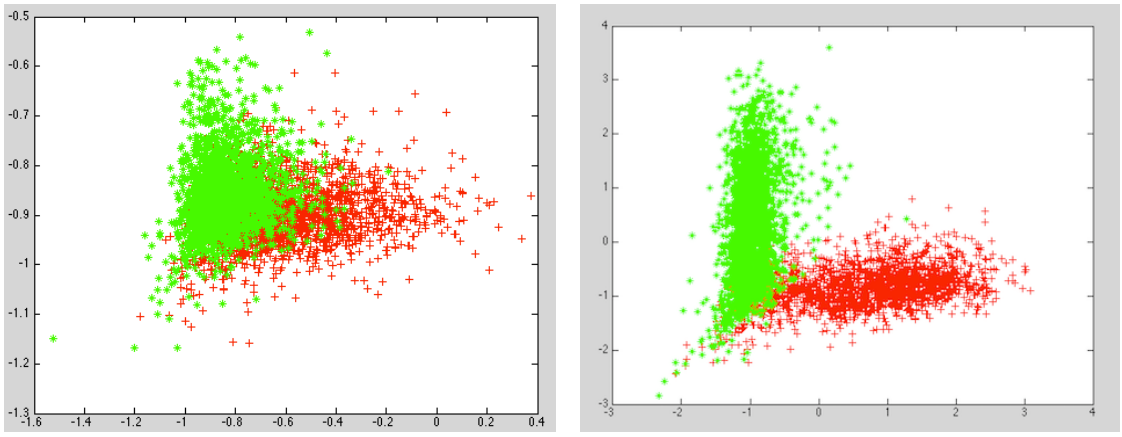
- We aim to classify dogs and cats based on the scores.
- We train SVMs with different kernel functions on training set scores and classify test set scores with the trained SVMs.

Usage of Asirra Dataset and Computation Cost

- Due to the large scoring computation time, which is 1000 image for 2 hour, we score 4000 training images (equally distributed in cat or dog category) and use them as the score training set.
- The training process is achieved on two Macbook Pro from me and Landu separately running cat model and dog model training. This takes around 10 hours.
- However big cost of computation the test set will incur, we have to at least run the head model processing on the 12,500 test images to submit our result to the Kaggle competition website to get an unbiased and justified classification result. This is achieved on two MBPs and one Ubuntu desktop for around 8 hours.
- Due to the relative poor performance of body DefPM, which shall be illustrated in latter section, we omit the processing of test set and used the training set result for an direct illustration.

Score Pattern Overview

- Red for Cat and Green for Dog.

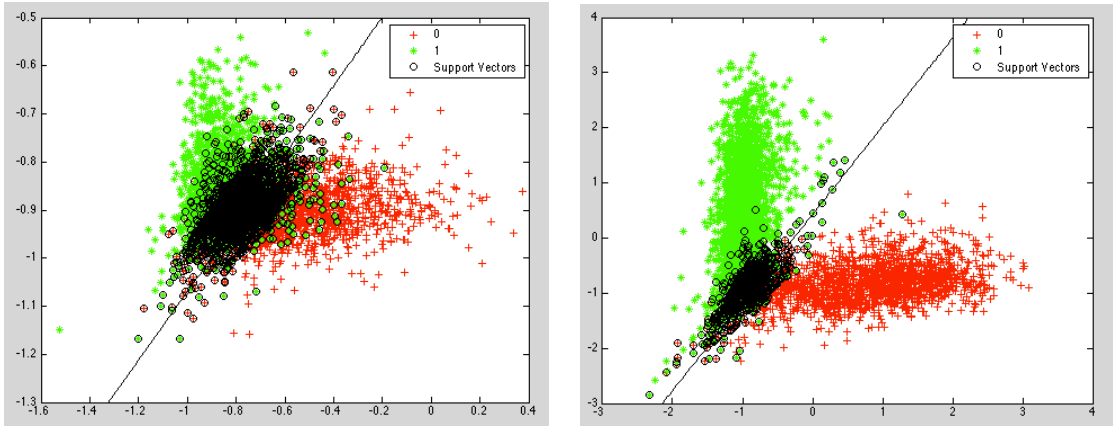


(a) Body DefPM (b) Head DefPM

Figure 4.1 Score Pattern Distribution

SVM Training Result

- We present the decision boundary for Linear SVM as intuitive illustration.
- Black colored samples are misclassified instances.
- Red "+" is Cat and Green "*" is Dog.
- The circled instances are support vectors.



(a) Body DefPM (b) Head DefPM

Figure 4.1 SVM Decision Boundary-Linear SVM

- We trained different SVM based on kernel types. And compare

their performance in the table below.

Table 4.1 SVM Training Result

Accuracy Kernel\	Body Model Train set	Head Model Train set	Head Model Leaderboad
Linear	0.7750	0.9177	0.9058
Polynomial	Coverge Fail	0.9167	0.9017
Gaussian	0.7782	0.9193	0.9061

ii. Score classifiers for BoW

In our training process for appearance-based Bow, we choose 6000 images in total (3000 for cats and 3000 for dogs). Then we use half of the dataset in training and another half in testing. The cat training images will be used as the negatives, and the dog images as the positives. The classifier is a standard Support Vector Machine (SVM) as well as 1-layer Neural Networks with 10 perceptrons. For SVM, we experiment with linear kernel function, and for the Neural Network, we used limited iterations (20 iterations) for a better convergence. Finally, the learnt classifiers are processed with the test images and look at the qualitative performance by using the classifier score to rank all the test images (shown in Fig. 3.3).

For linear SVM, we measure the performance quantitatively by computing the Precision-Recall curve [9], which is computed by varying the threshold on the classifier (from high to low) shown in following Fig. 4.3. The In order to assess the performance by a single number (rather than a curve), the Average Precision (AP, the area under the curve) is computed and shown in table 4.2. Meanwhile, we present the accuracy of the neural network together with its ROC curve in Fig. 4.4.

Table 4.2 Classification Result (BoW)

Classifier	Training Accuracy	Testing Accuracy
Linear SVM	0.6435	0.6410
Neural Net-10 perceptrons	0.8307	0.8193

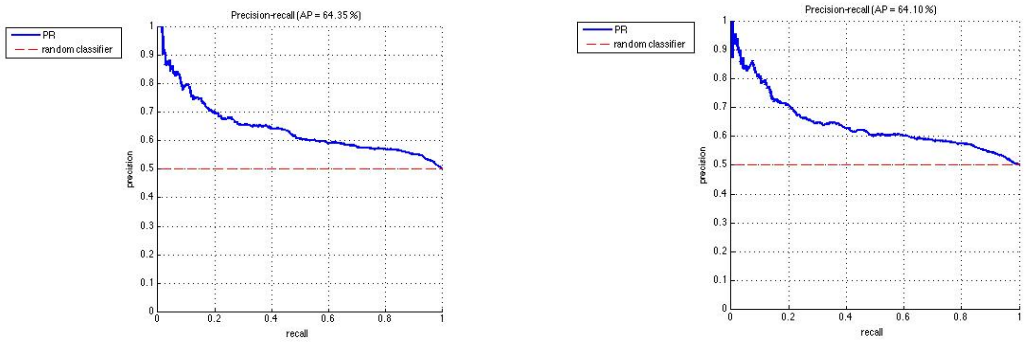


Figure 4.3 The Precision-Recall curve of Linear SVM

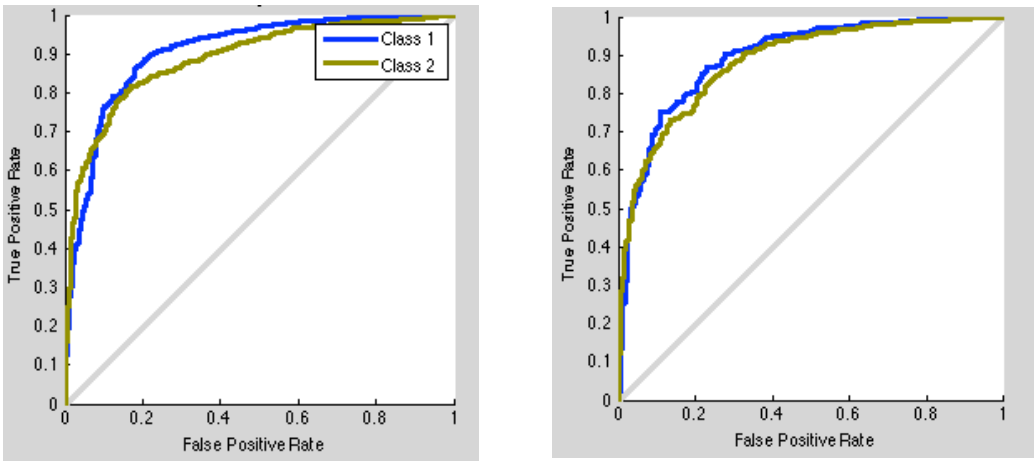


Figure 4.4 The ROC curve of Neural Net

V. Evaluations

We provide evaluation based on Deformable Part Model and Bag of Visual Words. We give separate insights into the shape based and appearance based models.

i. Evaluation of Deformable model.

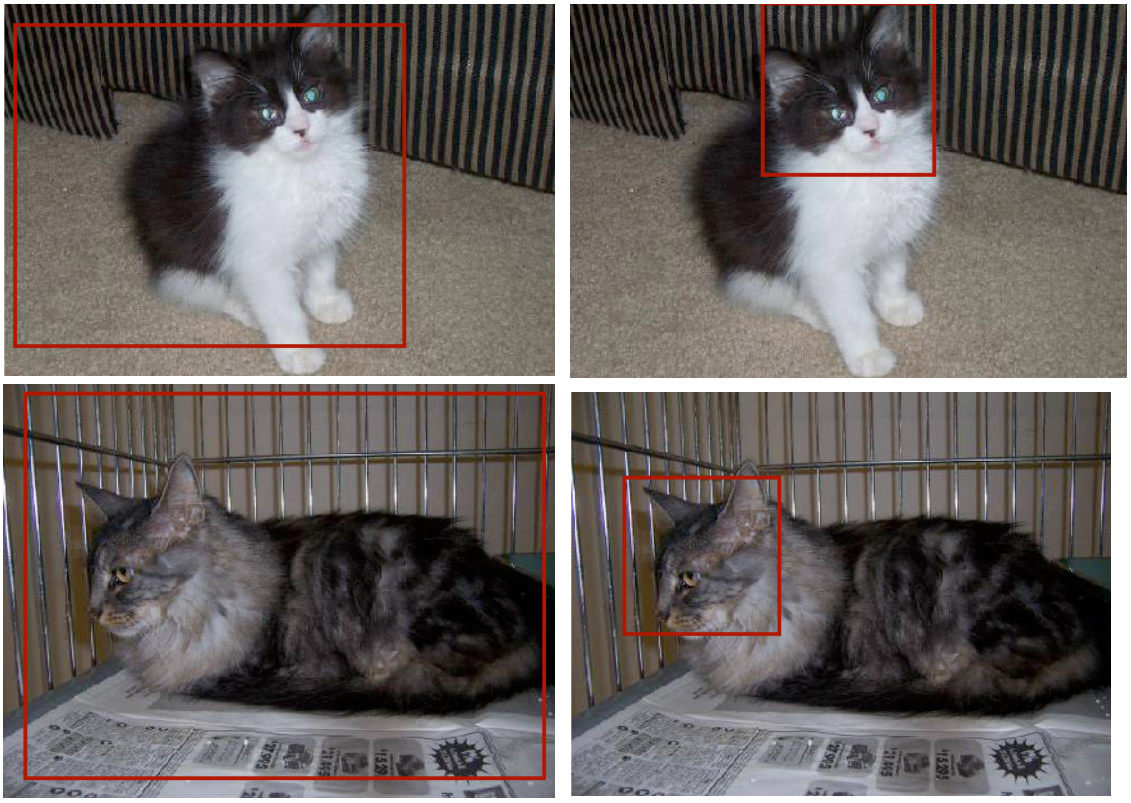
In this section, we evaluate the DefPM from perspectives of SVM kernels, body and head comparison and drawback analysis. Generally speaking, the Deformable Part Model is performing very well with head part, which achieves 90.6% accuracy on the Kaggle leaderboard. The Kaggle itself already provides a well evaluation of our work.

On SVM Kernel Functions

- Little difference between performances on kernel functions.
- The scores are well linear separable. Thus polynomial kernel works worst and linear SVM is a ideal choice for accuracy and convergence speed.
- RGB works the best. Though the gain is relative tiny.

Body DefPM vs Head DefPM

- Bodies are very deformable because of difference positions a pet can hold. Thus body is hard to capture exactly and the bounding box are including more noise than head box.
- Head are undeformable and stable with much less noise included in bounding box and scores.



- Noisy scores are the fatal drawback of body models, which make the score distribution much overlapped and hard to separate.

Drawback Analysis of Head DefPM

- Score is meaningful only when the head is correctly detected and bounded.
- The hard examples lie in two major categories: noisy pictures or hidden target parts.

- The above examples all make the achieved score noisy and making no sense. Thus misclassification is reasonable in these situations.

ii. Evaluation of BoW

Generally speaking, the neural networks did a good job of classifying the histogram features. Not only did neural net promises a better accuracy, but also a more smooth and concave ROC curve. Meanwhile, the linear SVM did not give a good accuracy neither did it produce a concave Precision Recall curve.

The lack of geometry in the appearance-based BoW model can potentially be either an advantage or a disadvantage. On one hand, without the relative geometry information, we get significant flexibility to encode only the occurrence of the appearance of the local patches on viewpoints and pose changes. However, on the other hand, the BoW will miss the important discriminating geometry factor between appearance features. In practice, features can provide some implicit geometric dependencies among the descriptors and one can achieve an intermediate representation of the relative geometry. Such appearance features are often extracted from overlaps of patches in an image. Furthermore, the background features may "pollute" the object's appearance when the BoW is extracted globally – foreground and background features are mixed together.

iii. Comparison

In the content of addressing Asirra dataset, the shape based Deformable part model clearly beat the appearance based BoW method. It has a gain of around 20% accuracy and largely reduced the dimensionality from 4000 which is the vocabulary size of BoW to the only 2 scores of cat response and dog response. With this strong feature construction model, only simple classifier can achieve amazing job.

However, this is not declaring that the shape based model is always better than the appearance models. Appearance models have much less processing time and straight forward theory. What's more, the DefPM always need extensive amount of images annotated with particular part to finish it filter training, which are hard to achieve without human effort. Meanwhile, the training process of DefPM is much more complex and cost orderly more time than appearance processing does.

VI. Conclusion and Future Work

We did dedicated work of implementing the DefPM as well as the BoW method. Then, we construct the raw image with the above models into new feature space and tried various classifiers to distinguish cat and dog based on them. Finally, with the more advanced and sensitive head deformable part model, we achieved 90.6% accuracy on Kaggle leaderboard. Evaluation and comparison is also provided according to our temporal results.

Our future work, if applicable, lies in trying more advanced appearance models as well as tuning the deformable part models. For instance, we may try to eliminate the background and extract the foreground of the image. We may also use the DefPM to locate head area and then applying the BoW method. Last but not least, the latent SVM training is still having work to do in order to achieve a faster convergence.

VII. Acknowledgement

We want to give our sincere acknowledgement to Prof. Kaleem Sidiqqi, who gives us many help on feature selection and reference recommendation. Most importantly, we appreciate his effort of giving great lectures and incurred our interest in computer vision. Meanwhile, we want to acknowledge Dr. Felzenszwalb who created the deformable part model and all the researchers contributed to Vlfeat community.

Finally, we want to say thank you to Morteza for his long effort as a nice TA and his dedicated help for our course work.

Reference

[1] Kaggle, http://www.kaggle.com/c/dogs-vs-cats.\

[2] Asirra, http://research.microsoft.com/en-us/um/redmond/projects/asirra/

[3] O. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. The Oxford-IIIT PET Dataset. http://www.robots.ox.ac.uk/˜vgg/data/pets/index.html, 2012.

[4] P. F. Felzenszwalb, R. B. Grishick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. PAMI, 2009.

[5] O. Parkhi, A. Vedaldi, and A. Zisserman. The truth about cats and dogs. In Proc. ICCV, 2011.

[6] Parkhi, Omkar M., et al. "Cats and dogs." Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In Proc. CVPR, 2005.

[8] G. Csurka, C. R. Dance, L. Dan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In Proc. ECCV Workshop on Stat. Learn. in Comp. Vision, 2004.

[9] Image Classification Practical, http://www.robots.ox.ac.uk/~vgg/share/practical-image-classificaiton.html

[10] Bags of Words, http://www.cs.utexas.edu/~grauman/courses/fall2009/papers/bag_of_visual_words.pdf

[11] VLFeat, http://www.vlfeat.org/

[12] Voc release 4.01, http://cs.brown.edu/~pff/latent-release4/