Recognizing Cats and Dogs Using Deformable Part Model

Chu Wang

Instructor: Joelle Pineau

McGill School of Computer Science

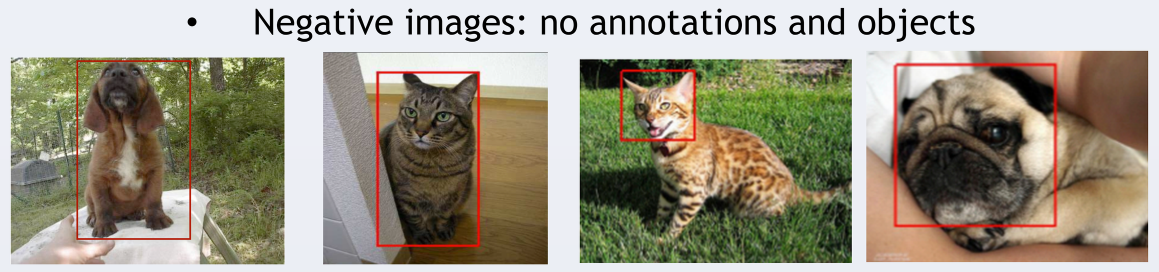
Abstract

Kaggle has raised up a competition to challenge Asirra's Pet CAPTCHA which contains dog and cat images and calls for state-of-the-art accuracy of recognizing cats and dogs from images with large variation in size and noise.

In this project, we participated the Kaggle competition and implemented a well known deformable part model[1] proposed by P F. Felzenswalb which is a detection system based on HOG features and is suitable for a large range of object class. We train the model on body and head annotations[2] to recognize cats and dogs in Asirra images. We achieved a very good accuracy of 90.5% with 10 over 70 ranking on Kaggle Leaderboard [4] and successfully challenged the Asirra CAPTCHA.

Introduction of Dataset

- **PASCAL VOC 2009**
  - Body annotations and original images
  - Negative images: no annotations and objects

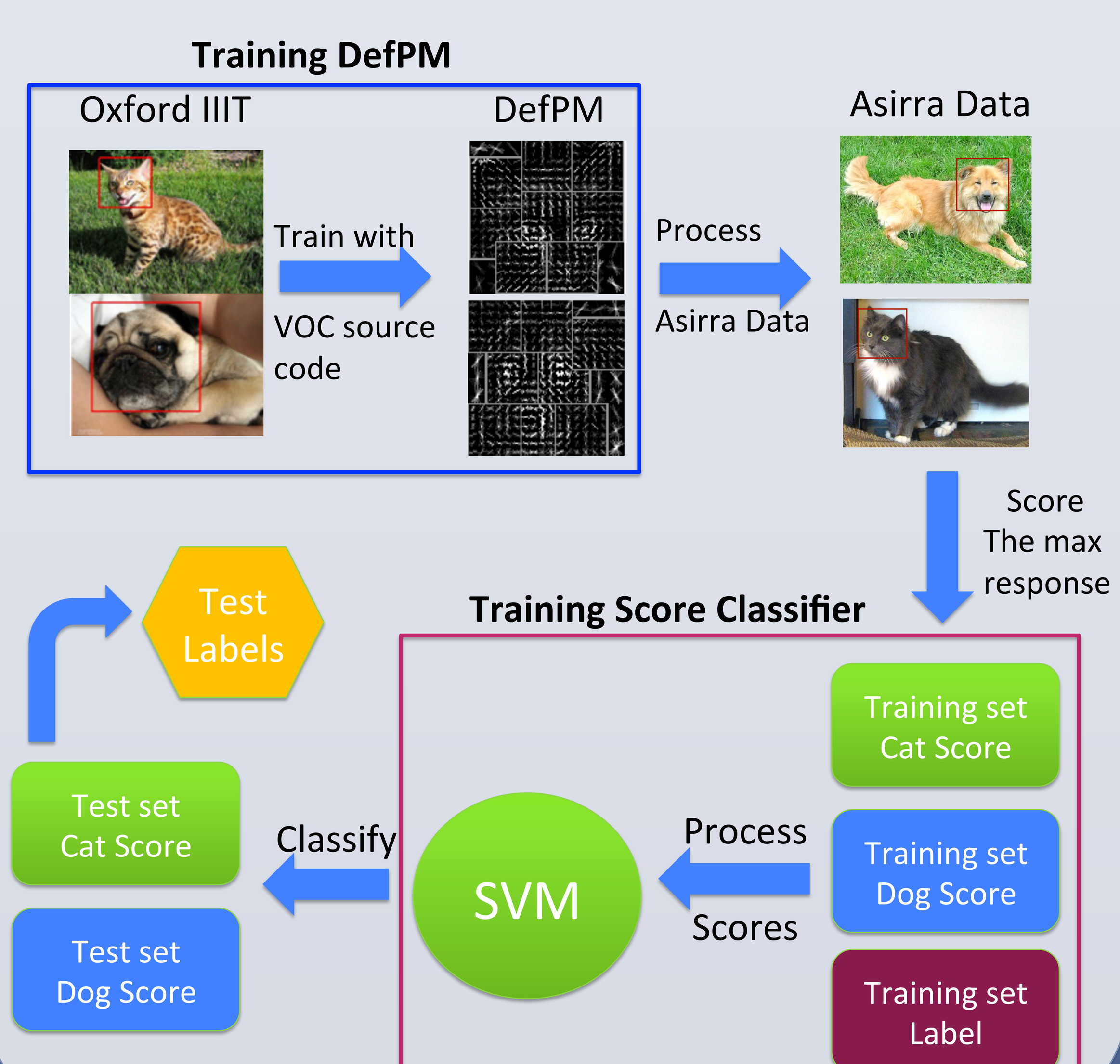


PASCAL Dataset — Oxford IIIT Pet Dataset

- **Oxford IIIT Pet Dataset [2][3]**
  - Breeds classification.
  - Offers head annotations
- **Asirra CAPTCHA Dataset**
  - Raw images with cats and dogs
  - 25000 training set/ 12500 testing set
  - Variations in size and bacground contents

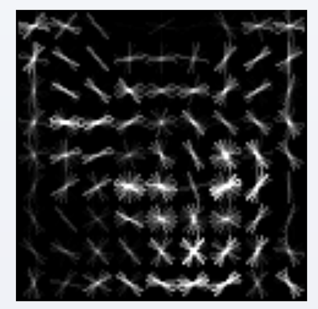
Solution Methodology

We demonstrate the training and processing workflow, here we use head DefPM[2][3]:



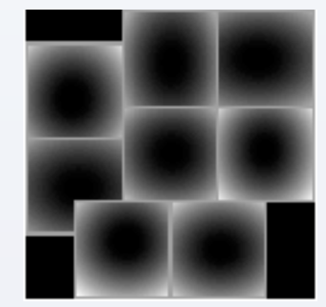
Score the Image: Feature Construction

The deformable part model (DefPM) [1] is a set of filters, among which there is one low resolution root filter and several high resolution part filters, and we display our head deformable part model as below.



Root filter

Part filter

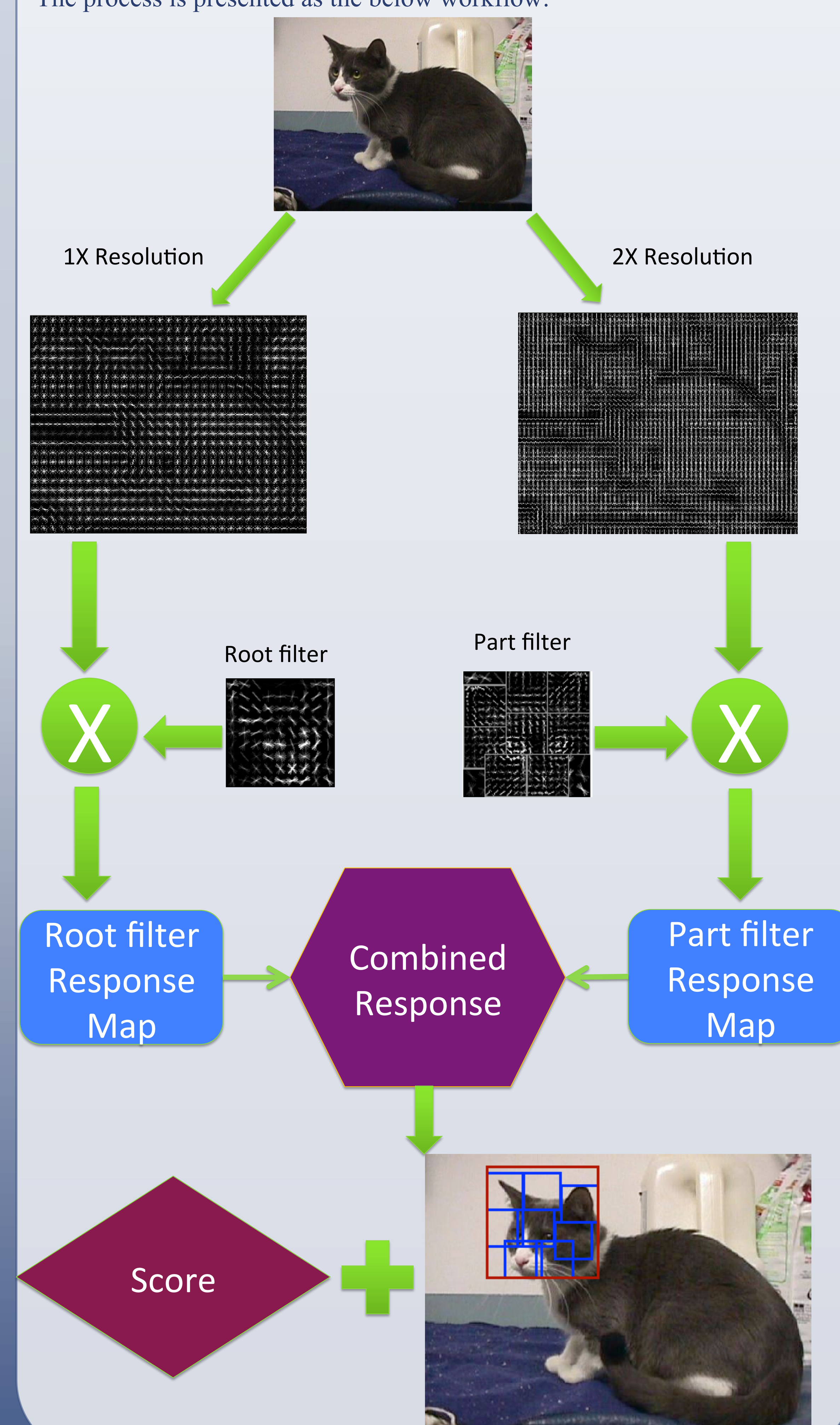


Part Positions

The scoring process of the model filters over a query image is done by the following steps:

- **Convolve the object root filter at the HOG feature map of original image and convolve the part filter with twice resolution.**
- **Combine the response of convolution into one response map.**
- **Locate the instance area where the response is highest and formulate boundary box.**
- **Record the max response as the score of object.**

The process is presented as the below workflow:



Train the Deformable Model[1]

Statement of Parameters

- The model has parameter vector β
- The feature space x
- Latent value: the bounding box locator vector $z \in Z(x)$
- The feature vector $\Phi(x,z)$
- The score of this image over model β :

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

Training datasets

- The PASCAL VOC dataset provides negative images with no target object within them. $N = \{J_1, J_2, \ldots, J_m\}$ and they are all labeled "-1".
- 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 $P = \{(I_1, B_1), \ldots, (I_n, B_n)\}$ and label them "1".

Latent SVM

- We have latent value z in our target function and we build a latent SVM

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

Open Source Code available[5]

Training Algorithm (Pseudo Code)

```
while(t<Num_iterations)
  for i=1 to n do
      Solve the latent value which covers 50% of bounding box.
          z_p(i) = arg max_{z∈Z(x_i)} β·Φ(I_i, z)
          st.   Box_β(z_i) ∩ B_i  ≥ 0.5
      Add to
          F_p(i) = (x_i, z_p(i), y_i)
  end
  for i=1 to m do
      Solve the latent value
          z_n(i) = arg max_{z∈Z(x_i)} β·Φ(J_i, z)
      Add to
          F_n(i) = (x_i, z_n(i), y_i)
  end
  Stochastic gradient
          β = gradient − descent(F_p ∪ F_n)
end
```

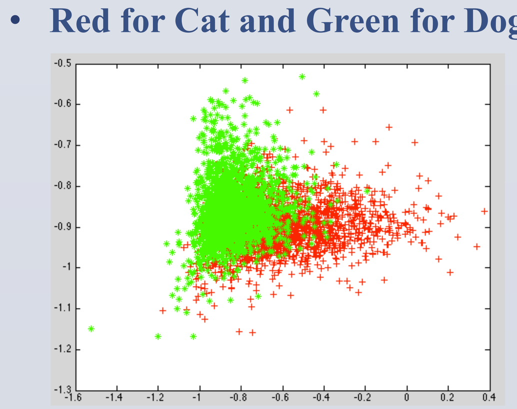
Train SVM score classifier

Statement of Task

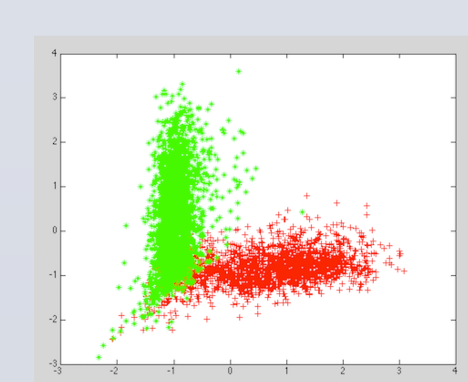
- **Due to the large scoring computation time, which is 1000 image for 2 hour , we score 4000 training images(equally distributed in cat or dog category) and use them as score training set.**
- **We need to classify dogs and cats based on the scores.**
- **We train SVMs on training set scores and classify test scores with the trained SVM.**

Score Pattern Overview

- **Red for Cat and Green for Dog.**



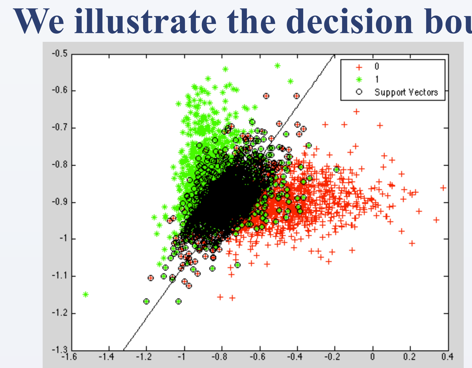
Body DefPM



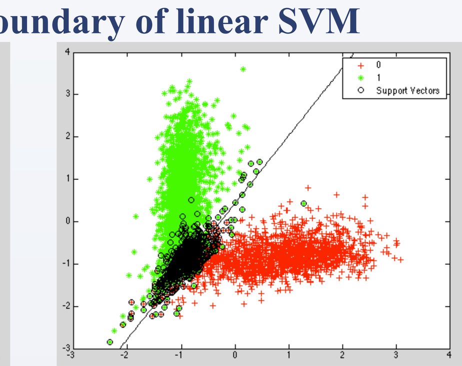
Head DefPM

SVM Training Result

- **We illustrate the decision boundary of linear SVM**



Body DefPM



Head DefPM

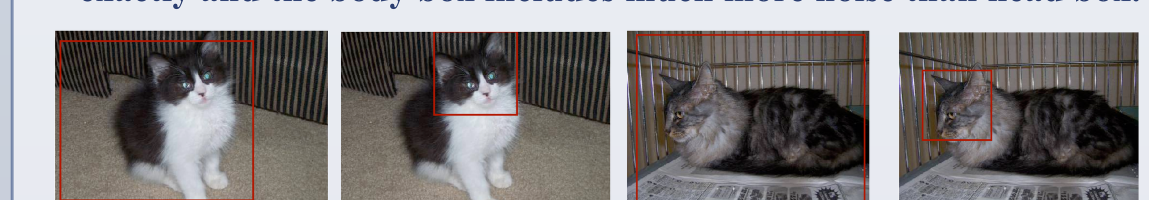
- **We trained different SVM based on kernel types. And compare their performance in the table below**

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

Conclusion and Evaluation

Evaluation:

- **We found that the body are very deformable because of different positions a pet can hold, which makes it hard to capture body exactly and the body box includes much more noise than head box.**



- **The main drawback of head DefPM lies in that the model succeeds only when the part is detected. If the picture is noisy or the part is somewhat hidden from camera, the scores becomes no sense.**

Conclusions:

- **Because head are relatively undeformable and stable, the head DefPM works much better than the body DefPM.**
- **The head DefPM is still vulnerable to pictures with noisy background which mainly contributes to the 10% error.**
- **Gaussian Kernel SVM works the best to separate the scores. Though the improvement is relatively tiny.**

Future Work:

- **Try appearance based classification method, focus more on color distribution to boost the performance of shape.**
- **Revise the training algorithm of DefPM to be more time efficient.**

REFERENCES

[1] Felzenszwalb, Pedro F., et al. "Object detection with discriminatively trained part-based models." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32.9 (2010): 1627-1645.

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

[3] Parkhi, Omkar M., et al. "The truth about cats and dogs." *Computer Vision (ICCV), 2011 IEEE International Conference on.* IEEE, 2011.

[4] Kaggle Dogs vs. Cats [Online]. Available: http://www.kaggle.com/c/dogs-vs-cats/leaderboard

[5] Voc relese 4.01 [Online]. Available: http://cs.brown.edu/~pff/latent-release4/