Classifying Images of Cats and Dogs Using Computer Vision

Shubham Saxena (ss4017@rit.edu) and Vishal Manghnani(vjm1952@rit.edu)

Department of Computer Science

Rochester Institute of Technology

1 Lomb Memorial Drive Rochester, NY 14623

30 November 2015

Abstract

In this project, we are focusing on the discriminating between objects in images. The idea is to base the discrimination on features of objects that give a clear separation. As it is a huge task to consider every type of object, we are concentrating on working with images of cats and dogs. Discriminating between the two is easy for the human eye but automating this for a computers to understand has more depth to it as we don't have true artificial intelligence yet. Cats and dogs look similar in a lot of aspects like their body structure. They have finer discriminating features when it is to be automated using a computer vision algorithm. Differentiating cats from dogs in images is a challenging task of Computer vision. In this paper we present the description of the previous work that has been done on this idea and techniques we have used for our model.

1 Introduction

Detection of all objects in an image using computer vision is a challenging task. We would have to account for countless number of objects. For each object a feature or set of features that define it must be considered. For objects that are similar there must be secondary features that can be used to distinguish between the similar objects. This is a computationally expensive task. The main problems with object detection are large intra-class variance (for eg. the dog breeds of Chihuahua and St. Bernard look drastically different), low inter-class variance of similar objects (for eg cats and dogs all have a head, two ears, 2 eyes, nose, mouth, body, four limbs, tail etc), varying shape and position of the object of interest and noise in the image. The illumination and cluttered backgrounds of the image also make detection of objects difficult and less

generalized. So we need to go one step at a time and focus only on smaller set of objects like cats and dogs. It is a difficult problem for computer vision to discriminate cats from dogs as the features like a cat's head and body shape are similar to those of other animal like dogs. To find the unique features from the image using computer vision is a good learning experience as it involves shape and texture detection. The Cats and dogs image database is available freely as a lot of people love to upload the photos of their pets. In the following sections, we will talk about the previous related work in this field. We will then move on to describing our approach, the challenges involved in it and how we try to solve them. We will talk about possible business applications of our problem statement. We will conclude with the discussion of results and future scope of the proposed model.

2 Related Work

In the paper The Truth about Cats and Dogs [12] Parkhi et al talk about capturing the images of animals by taking into consideration the shape and the appearance of fur and then segmenting the image of animal from its background. Different models are described namely

- Shape Model
- Appearance Model
- Automatic Segmentation

These deals with extracting features from the image that can be used to train the model. Each image is annotated to include the pixel level segmentation corresponding to foreground, background and bounding box around the head. Many images have homogeneous coloring which can be used to develop a method which outperforms the traditional methods.

In the paper [7] Machine learning attacks against the Asirra CAPTCHA the authors try to develop a technique that could be used to offensively or defensively exploit the capabilities of Asirra captacha so they create a test dataset to verify the classifier performance in classifying the Asirra captcha. They collected 13,000 images from publicly available Asirra database to train the SVM classifer. It extracts the color and texture features from the image. To extract color features HSV color model is used and the image is divided into a number of bands from which a boolean vector is generated based on different colors. Texture features are generated by dividing image into texture tiles and distance between the image and texture tiles are used to build a feature vector. The color and texture features are used in combination to train the SVM classifier.

In the paper by Kozakaya et al. [10] Cat face detection, authors have described the method to detect cat faces using Haar like features with AdaBoost and CoHOG with linear classifer and combination of both

gives a fast and efficient cat face detector. CoHOG descriptors use the sobel edge detector to calculate the gradient with orientation and cooccurence matrix is calculated with paired orientations giving the high description feature vector from which even a simple linear classifier can classify with high accuracy. In joint haar like features the image is represented by haar features and each value is quantized to binary values of 0 and 1 which is then fed to Adaboost to select the most discriminating features. Combining the haar like features and CoHog gives better performance than traditional methods.

Kaggle.com provides a platform for data science enthusiasts to work on real life problems. Along with the cats and dogs image dataset provided, there are also submissions of the teams that worked on this problem and the results that they achieved. Most of the teams have used CNN (Convolution Neural Networks).

3 Issues

There are several issues in the past which are still not fully solved. The dataset from Asirra contains the images which are in grayscale. These images contain other animals too and there are also a lot errors in the breed labels that are the main source of noise. To remove such issues each image needs to be labeled manually by considering different labeling techniques. Datasets from flickr.com [10] comes with already annotated data as the users label the images after uploading it. This data has some wrong labels. As the input data is wrong the classification cannot be done correctly. However the main issue is with the cat head detection as it contains large intra-class variation which is more than the human face and the cat head has more texture information than human face. So to detect the cat or dog face we need to take the texture and shape information together. Also since the majority of datasets available are annotated we need to worry about the face detection which can be done by using deformable shape model and texture information by using bag of visual words model[13]. The main task is to classify the breed which is not included in some annotations. But as the breed is dependent upon the texture information it can be exploited to classify breed as we are already taking texture features into consideration.

4 Challenges

Challenges faced in the past were mainly due to diversity of the images taken from different domains. The dataset in paper[13] was collected from different pets site like Petfinder, catster and dogster. The dateset consisted of 37 different breeds of cats and dogs which included 12 cat and 25 dog breeds. The main challenge was to deal with the noise in the image which was due to redundancy, poor illumination and in some images the pets were wearing the clothes due to which getting the texture and color information becomes difficult.

To resolve such discrepancies human annotators assigned labels manually and 200 images of each breed were finalized for training. Various features were extracted to segment body, face and other features.

In the cat head detection[15] the main challenge was that there is no perfect object detection algorithm which can be used to accurately detect the cat heads. Head detection is a challenging task due to large intra-class variation in the the cat head, Also images contain lot of noise because of varying pose, partial occlusion and cluttered background. To overcome these challenges the images require proper cleaning without affecting the cat head or shape which is not possible with primitive object detection algorithm.

5 Features

As we have already described some of the features in all the above section. We will sum up all the possible features that can be used in our project to improve its accuracy.

1. Color features are the most important and the easiest feature to extract. We can extract the color and convert it into the HSV model since it is more relevant to human visual system. Using color combination we can create a color feature vector.

2. Texture features are the most discriminating feature in making a difference between cat and dog. Since the texture is just the quantitative measures of intensity levels. This feature will also help to detect face.

3. Shape feature can be used to consider the different positions of the animal which gives us the overall structure of the pet.

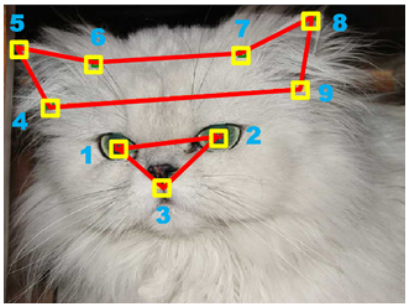
4. Edges of faces: edges of faces of both cats and dogs can be extracted and used for classification with relative easy implementation using FLD (Fisher Linear Discriminant).

5. Bag of visual words by extracting SIFT descriptors is another important feature to discriminate between multiple animals as the image might contain both cat and dog and some other animals too.

6. Object segmentation can be implemented by using Gaussian mixture model with Berkeley edge detector[13] along with grab cut segmentation. This is promising technique for extracting cat or dog from image.

7. Histogram can be used in combination with bag of visual words to create a diverse feature vector that can be implemented to detect head and body. [1]

Figure 1: Cat face features (source: google images)



1 - Left eye
2 - Right eye
3 - Mouth
4 - Left ear-1
5 - Left ear-2
6 - Left ear-3
7 - Right ear-1
8 - Right ear-2
9 - Right ear-3

6 Data Set

The data set we started working on was picked up from competitions on Kaggle.com.[11] The dataset is a a collection of 25,000 images of cats and dogs. We intended to use these to to train our classification model. With the images there was an associated .csv file that has the image identification number and class variable which signifies 1 = dog and 0 = cat. We also have a test data set which we will use to test the accuracy of our model. This dataset has been compiled by Microsoft research in collaboration with PetFinder.com. However the dataset was extremely noisy. The object of focus that are cats and dogs are not all usable. Many images were blurry and many had bad illumination. Most pictures have a large background with various objects. There are images in which people are holding the pets or have them wearing clothes. Many images have the cats and dogs oriented and looking away from the camera. As our capabilities and expertise of image processing are not very advanced we chose to get easier data. We reduced our data set to images of cats and dogs faces that were well illuminated, centered and not minimally skewed in orientation.

7 Ethical considerations

The images in our project has been selected from the Kaggle Data challenge which is publicly available and anonymous. Also there is no privacy issues as people generally upload their pets photos easily. However the main concern is the background information which might contain some sensitive data but we will be extracting and focusing only cats or dogs from the images and eliminate it's background. We are sure that no animals were harmed for the collection of these images.

8 Business Applications

Our project aims to distinguish between cats and dogs however it has a number of possible applications. An automated computer vision system to recognize and distinguish between cats and dogs have a huge market in the field of domestic pets. Pet owners face everyday situations that can be solved simply by using visually intelligent computer systems. For instance, while traveling out of their hometowns the pet owners have to leave their pets with a friend or family relation. The other option of looking for pet kennels or pet hotels can be very expensive and cumbersome. The pets usually have to be restricted to small enclosures and are mentally effected by such arrangements. Then there is the chance of having aggressive encounters with other animals. An automatic pet feeder could enable pet owners to leave their pets in the comfort and familiar environment of their home. CatFi.com is one instance of an automated cat feeder (and other care products). [2]. Assuming the pets are toilet trained they can be left at home alone for at least a period of a few days. But why does a pet feeder need to be able to visually recognize the pets? We can think of a few examples, for instance when a pet owner has more than one pet. Cats and dogs are usually given different diets. A visually capable automatic feeder would would supply different food materials to different animals. A more robust and trained model capable of distinguishing one dog from another dog would be even more useful, so that the same dog is not fed twice and/or given the correct amount suited for the specific dog considering the age etc.

Most pet owners would like their pets to have the freedom of independently going out into the backyard and come back on their own for cases like when they need to relieve themselves. Pet flaps built into the doors do allow the pets to enjoy this freedom however there are numerous incidences of stray animals like raccoons and skunks walking in to the house where they are unwanted and cause trouble. For this reason pet owners consider the restriction of their pets liberties to be the safer option. A pet gate keeper that would open only after recognizing the pet could help prevent unexpected entry of rodents and other stray animals.

Many Security alarm systems are configured to simply detect intrusion. These intrusions are often set off by the neighbor's pets and stray birds and animals which do not (usually) hold the risk of robbing houses! For these security alarms it is essential to have a low false positive rate.

In the unfortunate event of when a pet goes missing. We can use computer vision in combination with the surveillance systems to help find the missing pets and return them to their beloved owners.

Pet finder is a website for pet adoption and animal welfare work. It utilizes computer vision to process the images of the pets that users upload. Using computer vision they can perform automated labeling of the pets which can be later used for fast and accurate searching and displaying results for users.

We use Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA)

on the web to prevent automated bot attacks. A popular and often used CAPTCHA techniques used until 2014 was choosing cats from a set of 12 images given. This task is intuitive and easy for humans and an accuracy of 99.6% in under 30 seconds is observed. On the other hand making this task automated using computer programs (without using vision) is capable of 1 correct answer in every 54,000 attempts. [5] Asirra which used this CAPTCHA technique shut down in 2014 due to the capability of computer vision now being able pass this test with human like accuracy. [14]

9 Challenges

Our initial approach was to first convert the image data into a set of human readable nominal, ordinal, ratio or interval data representing the features which would be used by our model to make the decision of whether it is an image of a cat or that of a dog. Finding these features were the most crucial part of our project as they would be the basis of the classification of the input images.

In cases where we have cats and dogs both we would have a conflict in the classification. This could be resolved if we allow the option of both existing in the same image. So our classification would change to 1 = dog, 0 = cat and 2 = both cat and dog.

Dealing with the data and converting it to usable features is the first and the biggest challenge we faced. We intended to convert the images into a set of extracted features that can be parsed by our classification model and based on these extracted features, give us a the classification of the image. This project seemed to have challenges at every step! The essence of this project can be broken down into two broader steps. First, being able to detect cats and dogs in the image and then say with some confidence which one was detected. The data consists of images. Which means a model needed to read image data and extract information which can be used as features. How this extraction is to be done and which elements can become the features needed a good understanding of computer vision concepts. For example which filters and detection methods would work best to detect the figures of the cats and dogs in the images and whether those features will be strong enough to allow our model to distinguish between a cat and a dog. The distinguishing features will require deeper knowledge and will allow narrower margin for error for the classifier. In figure 1. referred from [5], show how a close match or similar looking can be difficult if we look at only broad features like color and shape. A metric of the face shape and structure may be needed for a programmed algorithm to characterize and distinguish a cat from a dog.

Orientation of the faces of cats and dogs should be kept in consideration. In order to do this we required a method that would be unaffected by the animal's posture or different head orientations in the image.

Another difficult hurdle was to take into account the numerous breeds of cats and dogs that would

Figure 2: Close match of a cat and dog.

increase the variety of features to be considered. Some of the dog features overlap into the cat domain and vice-versa. Our hope was to be able to find similar features of cats and dogs that are not dependent and unchanged with the change of breeds along with being unique to cats or dogs.

10 Our Approach

As our scope is to distinguish between cats and dogs, for simplicity we are proceeded with the assumption that we do not have to deal with a very large back ground.

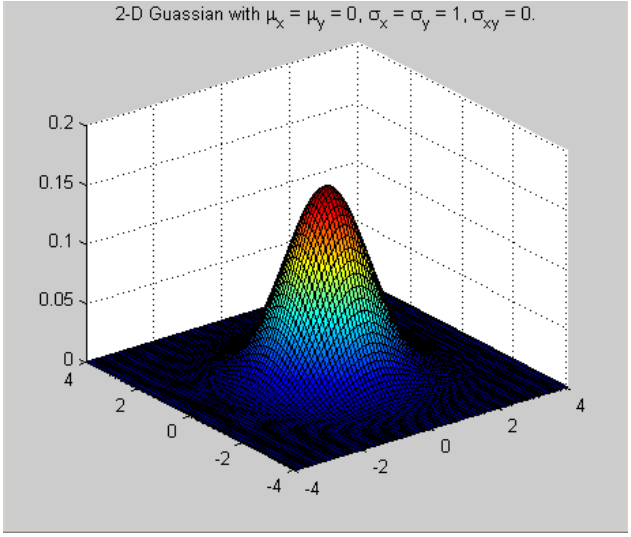
In the image anything except the dog and cat faces is considered as noise. The data set from kaggle are real images. They have been clicked by people for their personal collection and were not concerned about computer vision. Some images are of different scale and rotation. They are not all well illuminated. Some images have cats and dogs looking away from the camera. Some images have pet owners holding the cats and dogs. In some there are multiple cats and dogs while some images are dark or blurry. To keep our focus on the problem we have rejected all of these images. And made our own training data set with specific requirements:

1. Only front facing well illuminated cat dog images.
2. Centered faces of the cats and dogs. Cropped images as required.
3. Convert them and scale them to a resolution of 256 x 256 pixels.

We have our images converted to gray scale. Color is not being considered as a feature. Once we have the images, they are processed according to our model requirements stated above. We then apply edge filters to the image to get the strongest edges. We tried Canny and Laplacian of Gaussian (LoG) filters. We decided to go with LoG filter as it gives us finer edges. The Gaussian filter has been used to smoothen the image before edge detection. These edges are the primary features considered to classify the images under cats or dogs. It is a simple approach but gives us surprisingly promising results. We learned this approach from the paper Image Recognition for Cats and Dogs by Chung et al. [3]

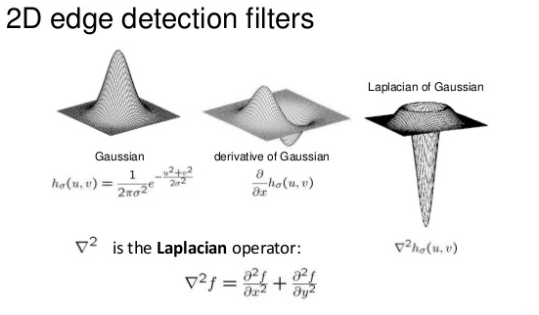
A brief description of the filters used and other used terminology: Gaussian Filter: The Gaussian is a smoothing filter. It is a 2-D convolution operator that blurs the images and helps remove details and noise. In essence it is a mean filter, but its kernel is different and represents a Gaussian hump or bell shape. Its purpose is to be used as a point-spread function by using convolving it with the image. We have used the following Gaussian filter : [121;242;121]

Figure 3: A 4x4 Gaussian Filter example [6]



LoG filter: The Laplacian of Gaussian Filter. This filter is a 2-D isotropic measure of the 2nd spatial derivative of an image. Using this filter we can highlight rapid intensity changes in the image. This helps us look for edges in the image. Usually the Laplacian is used on an image that has been smoothed first with a Gaussian smoothing filter or something similar so that we reduce the edges being detected near noisy regions.

Figure 4: A Laplacian of Gaussian Filter example [4]

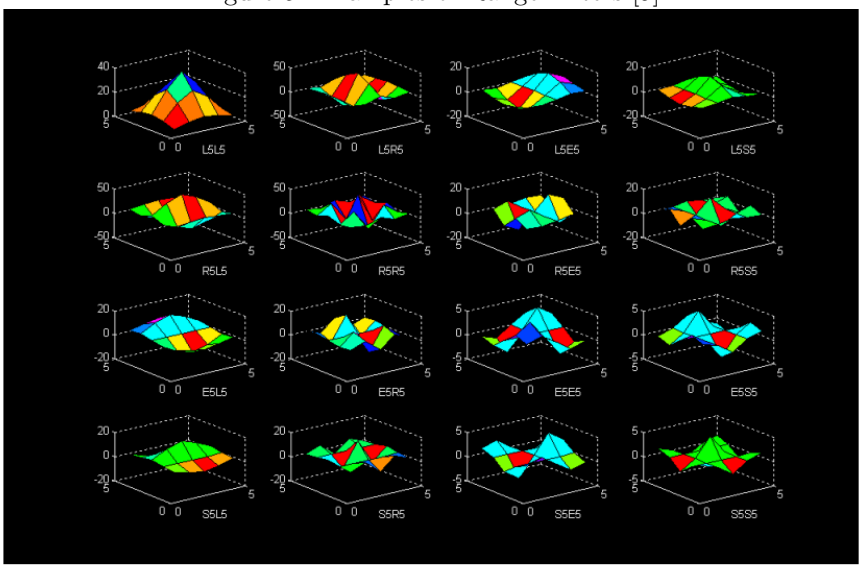


Texture: Texture analysis is the categorizing or characterization of areas in an image with respect to their texture characteristics. Texture can be described to be rough, smooth, bumpy, uneven, similar frequency of

patterns like liens etc, and can be represented by a function of the spatial variation in pixel intensities. We can use Texture analysis to evaluate texture boundaries. This is termed as texture segmentation. Texture analysis can be helpful when objects in an image are more characterized by their texture than by intensity, in this case a cats fur and dogs fur can be used to segment it out from the background.

A function in Matlab is Rangefilt() which returns the range values of each pixel in its set neighborhood (in our case 3x3). The range = (max value - min value) where the max value is the maximum intensity and the min value is the minimum intensity in the neighborhood of the pixel in consideration.

Figure 5: Examples of Range Filters [9]



Thomas Kinsman

The main aim of our classification is to find a low dimensional representation of our images. This is because we have each image represented by 256x256 pixels which are all features. So we have 256 x 256 = 65,536 dimensional measurements. We arrange all of these dimensions as the attributes of our image. We create a data matrix with each row being an image. We then perform PCA (Principal Component Analysis) using SVD (Singular Vector Decomposition). SVD breaks the data matrix into three components U, S and V. U is the weight eigen vector, S is the eigen values and V is the orthogonal eigen vector. SVD gives us the variance of the images from their mean. [8]

Our Classifier does 3 basic tasks:

1. Detect edges. First a Gaussian filter is applied to the image which allows smooth transition between intensities. This makes edge detection easier and gives better results. The Laplacian filter is a sharpening filter and it allows us to locate the change in the intensities. The change in the intensities are the edges.

2. Dimension Reduction. Too many dimensions will make the working very slow. Thus we have used SVD as an alternate method for faster performance. SVD greatly reduces the information that we work on.

3. Perform PCA with SVD to reduce the dimensions.

4. Calculate the threshold for the classifier.

5. Classify the test images with the help of the calculated threshold. [3]

Given below in figure 6 are the PCA images of the first four dataset features. We can see that there are clear cat features visible. So our classifier checks how close the test image is to these features. We observe that the mean image has visible cat features (figure 7). This implies cats will be strongly classified. Images that are not classified as cats will be dogs. Using PCA we can find the optimal projection direction where the two classes can be separable which will have large inter-class variance and small intra-class variance. Once we have the optimal projection direction we can project our data points onto this direction which will easily classify the two classes. To project the points we use FLD Fisher Leaner Discriminant. Once we perform FLD we get the weights and the threshold values which are used on the test images. The test images are projected onto the same FLD line and based on the threshold Fisher Leaner Discriminant description:

Figure 6: 1st four PCA of data

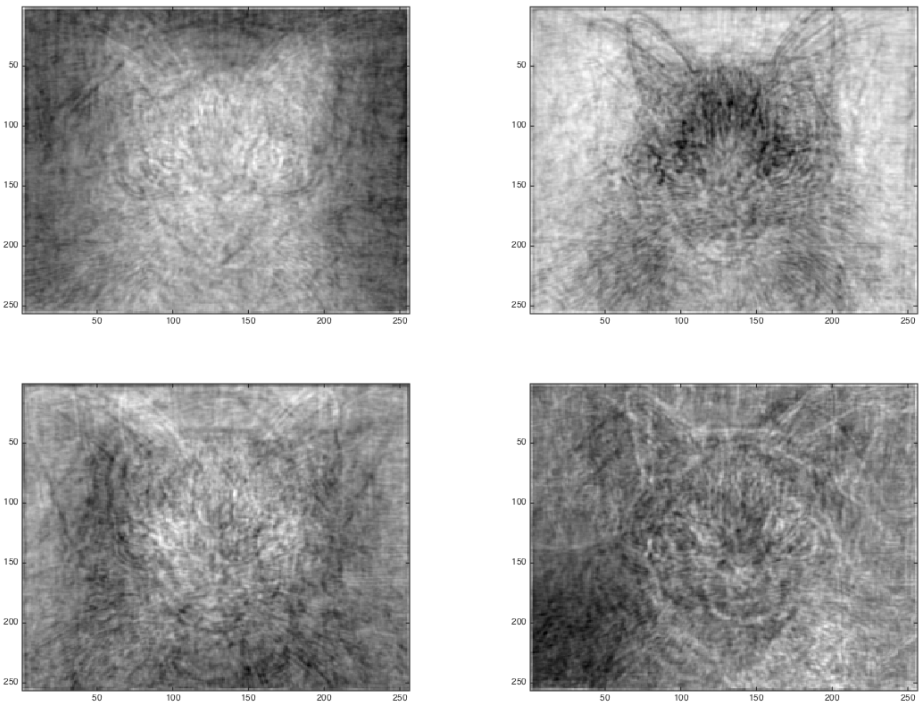


Figure 7: Mean image

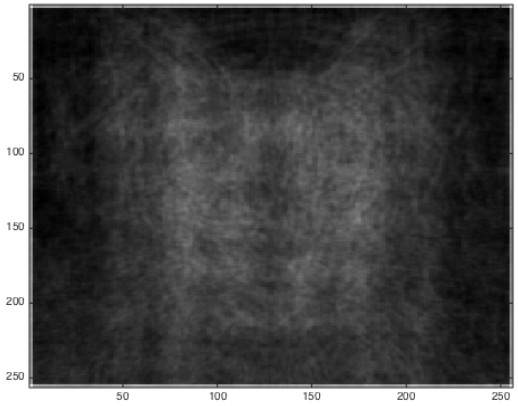
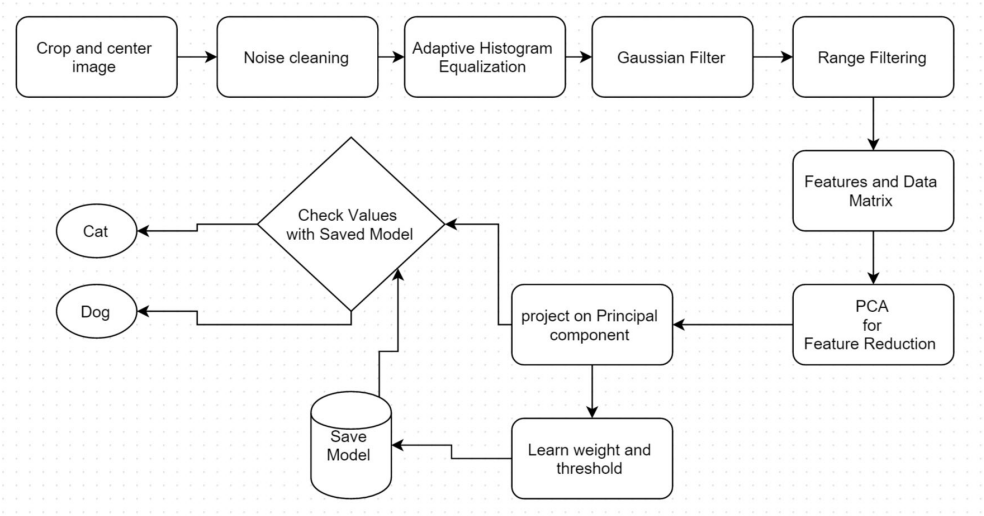


Figure 8: FlowChart



Methods that are being considered or were considered: Basing the classifier model on different features. like

1. shape/edges. -giving us results. We have to check how good.
2. Textures (Range of pixel values).
3. Haar Features: since cats and dogs are not very similar in facial features compared to humans we need new or different haar features. This is a very challenging task.
4. SIFT (Scale-invariant feature transform) : we will try this if time allows.
5. Deformable Shape model. This is a very difficult approach. Implementation will require much deeper computer vision knowledge and time to implement.

10.1 Results

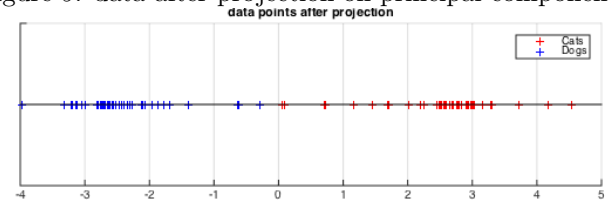
We trained our model on 50 cat and 50 dog images. We then tested the data with 30 fresh images consisting of 15 cats and 15 dogs. The results of the classification of the test images was as follows:

Confusion Matrix		
	Dog	Cat
Dog	14	1
Cat	1	14

As we can see from the confusion matrix there are 14 dogs and 14 cats correctly classified. The accuracy was $28/30 = 93.33\%$.

Figure 9 shows the points after projected onto principal component. There are overlapping points due to some noise and similar features in the images.

Figure 9: data after projection on principal component



False classifications: In some images of dogs which have their ears perked up like in the case of the husky shown in this image, the classifier gives the result as "Cat". This is because the shape is very close to that of cats which confuses the classifier to classify a husky as a cat.

Another case where we observed miss classification was when the dogs face fur variation and texture nature was similar to that of cats.

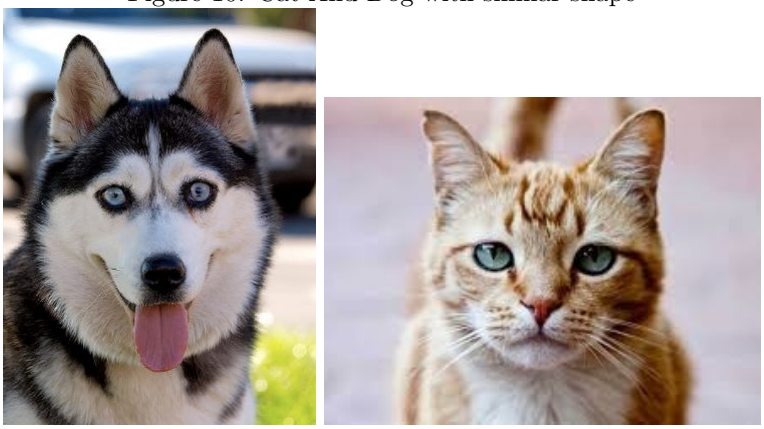
We expect if we could add a large number of images, our model would train better and give higher accuracy.

We get good results with dogs which do not have cat like ears like a Beagle, Labrador, Golden retriever etc.

10.2 Future Scope

We intended to take a reference image, which will have the appropriate rotation and scaling for our model. We wanted to then align all the images with respect to this reference image. For a more general model that accepts images with background noise, we wanted to attempt to make a face matching filter for cats and dogs and use this to eliminate background noise. As a result we would get a sub image with only the cat or

Figure 10: Cat And Dog with similar shape



dog faces. If we can achieve this, our model will be able to work only on the features of cats and dogs in the image by automatically ignoring the background noise. We are also considering application of histogram equalization to adjust the contrast of all the image.

After simply differentiating between cats and dogs, we can move onto more complex and specific problems. To further increase the accuracy of the cat dog classifier we could use eyes as a feature. Cats have vertical pupils while dogs have circular pupils. This would involve image segmentation which would first get us the face of the cat or dog then look for the eyes and check the characteristics of the pupils to give the final classification. This suggestion was given to us by our project mentor, Prof. Kinsman.

Another approach would be to look at the structure of the face. The ratio of the distances between the eyes and ears and eyes and nose (possibly nose and ears) are a good feature that can be utilized to differentiate between cats and dogs. This is what the human brain uses as a model to recognize objects, ie. it learns from repeated shape and ratio of distances between key features of an object. The challenge with this approach is that the orientation of the faces would need to be accounted for. Also a 3D mapping of the 2D images may be needed. For instance: most dogs have snouts. This makes the distance between the nose and the eyes longer for dogs while cats have flatter faces. In a 2D image when a dog has its nose pointing straight ahead the snout's depth is lost (for an automated system which looks at pixels) and the distance between the eyes and nose may become the same.

The next step could be the addition of intra-class classification/clustering of breeds. Dogs and cats were domesticated thousands of years ago and have been selectively bred for various purposes. Trying to figure out which breed of cats or dogs would focus deeper on finer characteristics like the finer structure of the face, body size, fur texture and color etc. This would be useful for applications like pet finder, which could give us specific search results. Example of different cat breeds and dog breeds are shown here.

As humans age their facial and body characteristics change with time. Similarly with cats and dogs aging

changes a lot of features. Fur loss and whitening of face hair are the clearest indicators. It is advantageous for automatic feeding systems to know the age of pets as their health and associated diets need to change. People who are not extensively familiar with cats and dogs find it difficult to tell the age of dogs by looking at them. This system if robust enough can be used by vets and animal control agencies to help take better care of the animals.

The scope of the project can be expanded to include more animals. Starting with domestic animals and moving to wild animals and birds. Maybe even aquatic life etc. Wild life photography is presently done manually or by cameras set up in remote places that need to capture months of footage to be able to give just a few seconds of interesting animal images and videos. This can be enhanced if we could make drones which are capable of animal recognition. We could then send these drones covertly into the habitat of some animals where it is difficult for humans to reach or capture their behavior without disturbing the wild life or influencing their actions by our presence. It would change the face of nature and wildlife research organizations like National Geographic, Discovery Channel and Animal Planet.

10.3 Acknowledgments

We would like to give a special thank you to Prof. Kinsman for his guidance and help. This project has been done in co-ordination with Akshai Prabhu. Our approach has been learned from the paper [3].

References

[1] O. Blog. cat face detection. `http://www.catfi.com`.

[2] CatFi.com. Pet cats care products. copyright 2015 zillians inc. `http://www.catfi.com`.

[3] H. J. Chung and M. N. Tran. Image recognition for cats and dogs. 2010.

[4] F. L. B. E. Detection. Laplaccian of gaussian filter for edge detection.

[5] J. Elson, J. R. Douceur, J. Howell, and J. Saul. Asirra: a captcha that exploits interest-aligned manual image categorization. In *ACM Conference on Computer and Communications Security*, pages 366–374, 2007.

[6] T. .-D. G. Filter. Gaussian filter.

[7] P. Golle. Machine learning attacks against the asirra captcha. In *Proceedings of the 15th ACM conference on Computer and communications security*, pages 535–542. ACM, 2008.

[8] A. Ihler. principal components analysis (pca) and the singular value decomposition (svd).

[9] T. Kinsman. Texture - range filters.

[10] T. Kozakaya, S. Ito, S. Kubota, and O. Yamaguchi. Cat face detection with two heterogeneous features. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 1213–1216. IEEE, 2009.

[11] Microsoft-Research. Cats and dogs. `https://www.kaggle.com/c/dogs-vs-cats/data` viahttp://www.kaggle.com.

[12] O. M. Parkhi, A. Vedaldi, C. Jawahar, and A. Zisserman. The truth about cats and dogs. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1427–1434. IEEE, 2011.

[13] 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.

[14] A. P. Shubham Saxena. Distinguishing images of cats and dogs using data analytics. Rochester Institute of Technology, 2015.

[15] W. Zhang, J. Sun, and X. Tang. Cat head detection-how to effectively exploit shape and texture features. In *Computer Vision–ECCV 2008*, pages 802–816. Springer, 2008.