# Final report

## Introduction

The topic of our final project involves classification between cats and dogs. This is a classical image classification problem in Computer Vision which mainly involves identifying certain features which can help us in clearly distinguishing between the two species.

The initial task at hand was to obtain a dataset which consisted of sufficient images to train the model. We obtained the dataset from Kaggel. The dataset comprised of 12500 images of each for both cats and dogs, along with more 12500 test images. We didn't however use all of those images for the classification task because of the computation time it demands.

We used the following methods to classify the images

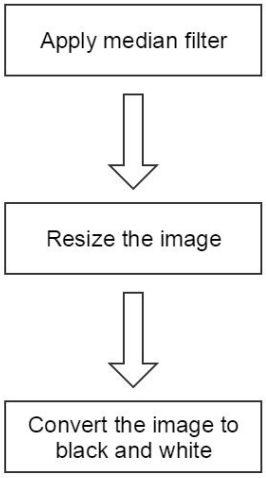
- Bag of Words
- Region Growing
- Convolutional Neural networks

## Classification based on Region Growing

In this technique we identify a feature which is unique to a cat or a dog and based on that key feature we proceed with the region growing. In our project we have chosen the **ears** as the distinguishing feature between cats and dogs. On close observation we see that the ears of cats are upright and triangular in shape but in case of dogs most of them have the ears towards their face and the ears are not upright as in cats. The procedure which was followed in applying the region growing technique is as follows:

- **Image Preprocessing:**

  In this step we apply a median filter in order to reduce the background noise in the image. This was followed by resizing the image to a fixed dimension to ensure uniformity and finally converting the image to black and white.




- **Region Growing:**

Once preprocessing is done we proceed to region growing. On the black and white image obtained we extract only the top portion of the image which consists of only the ear part of the cat or dog. Then we count the total number of black and white pixels from the extracted image. As the ears are our region of interest and only the ear portion of the cat or dog is taken into consideration the pixels which represent those areas will be smaller in number when compared to the background pixels. On taking the number count the count of pixels which are smaller in number constitute our region of interest or the foreground image and the other constitutes the background. For instance if the number of white pixels are 300 and the number of black pixels are 1000 then the white pixels constitute our region of interest and the black pixels can be neglected. Next we iterate from the first pixel in the extracted portion of image till we encounter a white pixel. Once we encounter a white pixel we check its neighbors to see if their pixel color is white. We also encounter the location of the first white pixel which was encountered and push it into a list. If one of its neighbors is also a white pixel we traverse to its neighbor, record its position and add it to the list. We proceed in the same manner till a point where a white pixel does not have any other white pixels as its neighbors. Then we can safely say that all the pixels in that list belong to one region. Then we again traverse from the next pixel after the first white pixel which was encountered. After traversing the entire image we count the total number of regions. On the basis of a threshold which is the region count we determine if its a cat or a dog. Since cats have both ears visible in the image the minimum region count will be 2. From the images we have tested dog images have a maximum region count of 2. Thus our program classifies an image which has a region

count greater than or equal to 2 as a cat and images having a region count less than 2 as a dog.

**Results**

This algorithm was run on 50 images of both cats and dogs and the classification accuracy was about 65%. When run for around 500 images the accuracy decreased to about 53%.

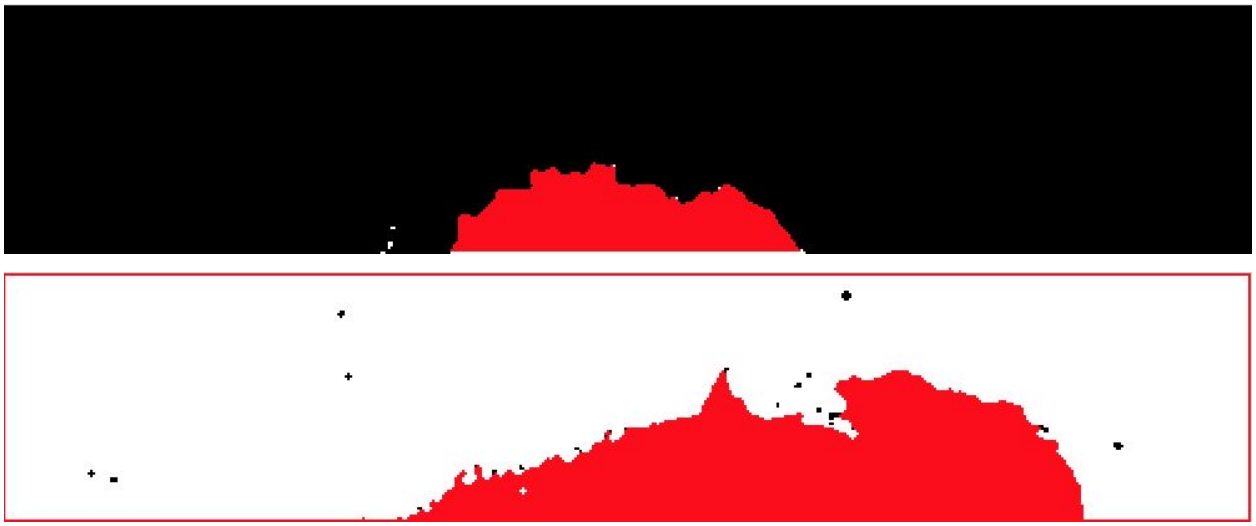
**Drawbacks**

In this approach only a single feature was taken into consideration which is the ears. Taking a set of features might have provided a better result as we are banking on the values of several features to classify as a dog or a cat. Some of the other features which could have been considered were the shape of the nose and the presence of whiskers in cats but not in dogs.Another drawback is dogs the ears of dogs differ from one breed to another so this algorithm may work well for a particular set of breeds and may not provide good results for others.

**Procedure to run**

python fp.py. Please ensure to have the PIL library.

Some of the screenshots are shown below for successful cases and failed cases.



Failed cases

## Bag of Words Model

In Bag of Words model we first develop a visual vocabulary and

***Build a visual vocabulary***

1) First we resize all the images into either 40*40.
2) We first get SIFT descriptors for all each images in the train folder.
3) Then cluster all the 128D SIFT descriptors into 100 clusters using K-Means algorithm. We used kmeans function from opencv for this operation.
4) We save the centers of these 100 clusters. These represent the visual vocabulary for our model.

***Represent each image as a histogram of the visual vocabulary.***

1) First we calculate the SIFT descriptors for the image.
2) Then look for the nearest visual word for each descriptor and generate a histogram for each image. We represent histogram as a vector of integers.

Use SVM to train the vectors containing histograms of each image. For this we used SVM Multiclass. Then Use the trained model to classify any new image given.

**Results:**

We trained 1000 images of cats and dogs, and saw an accuracy of 67% when tested for 100 random images

**How to run:**
Go to the bag of words folder and run the command make to compile the code.
To train the images use ./bow train bow
To test use ./bow test bow

**Note:** We only uploaded about 200 images for training and 30 testing images in github for you to be able to run and test code, as images consume lot of data. It will only take a minute to train and test with this much data.

## Convolutional Neural Networks

We tried to use caffe package to implement LeNet architecture. However we are not able to finish the modeling by the end.