B.Tech. Project Report

IT-413

on

Image Spam Detection Using Naïve Bayes Algorithm

BY

LAKSHAY HURRIA (1140444)

SATYENDER KUMAR (1140887)

MUJAHID OMER (2140003)

Under the Supervision of

Miss Mamtesh Nadiyan, Assistant Professor

DEPARTMENT OF COMPUTER ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
KURUKSHETRA – 136119, HARYANA (INDIA)
July – Dec, 2017

CERTIFICATE

I hereby certify that the work which is being presented in this B.Tech. Minor Project (IT-413) report entitled **“Image Spam Detection using Naïve Bayes Algorithm”,** in partial fulfilment of the requirements for the award of the **Bachelor of Technology in Information Technology** is an authentic record of my own work carried out during a period from July 2017 to December 2017 under the supervision of **Miss Mamtesh Nadiyan**, Computer Engineering Department.

The matter presented in this project report has not been submitted for the award of any other degree elsewhere.

Signature of Candidate

LAKSHAY HURRIA (1140444)

SATYENDER KUMAR (1140887)

MUJAHID OMER (2140003)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Date: 14th December, 2017

Signature of Supervisor

Miss Mamtesh Nadiyan

Assistant Professor

TABLE OF CONTENTS

ABSTRACT

Traditional text-based spam filtering is rapidly being replaced by Image spam. This is a sort of spam which includes the spam content in graphical form and has been the cause for many inconveniences online in the recent years. Now-a-days, we have image content analysis technologies which can be used to successfully detect these sort of spam. The previous attempts of spam detection only focuses on filtering the image spam on the client side. This can be problematic as both the server-side filtering and client-side detection will face difficulty to effectively mitigate image spam. Therefore, this project proposes a better desirable solution where-in the positive sparsity induced alike technique on the server side where-in the analysis of spam images takes place and the attack activities and events of spammer are filtered. While on the client side, the images are classified with maximum possible accuracy with the principle of active learning. The spam is filtered and blocked from the beginning after thorough analysis in the server-side. The automatic learner will help the users filter those spams which have made its way through the server side filter. For this project, experiments were conducted for various image spam collected online from e-mail which will demonstrate the workflow and provide a solution.

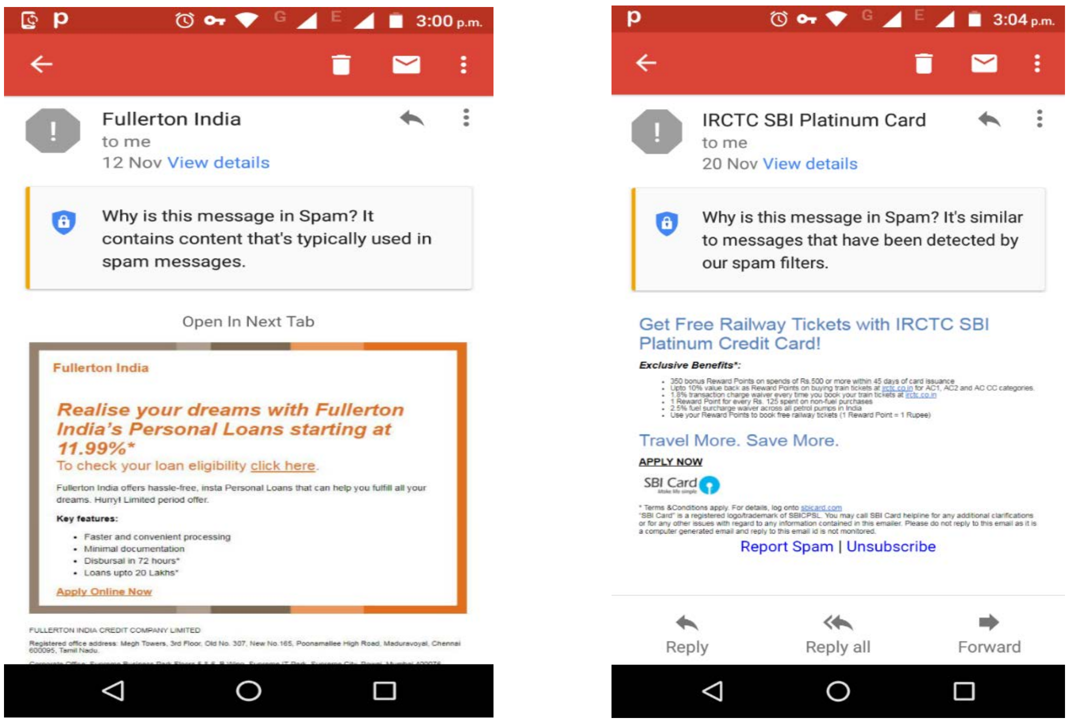
I. INTRODUCTION

We know that in today's world Internet is used very much. Today email use is very effective in communication. Emails are used to communicate at every phase whether it is professional or home use. So with the increasing use of emails hackers try to hack your personal information using spam in your email. Every day you get emails from the topic in which you are not interested. Like a student can be interested in academic information or study information but he is not interested in other types of information. A businessman can be interested in financial related information. So the information is relevant or not is purely dependent on individual. But we face many emails regarding the topics in which we are not interested. These types of mails can be spam or ham depending on individual interest. So we are concerning about only spam. This spam message is that message which is used to get the personal information of the user. So to get rid of image spam we need to analyze the image and after using machine learning algorithm like C4.5 decision tree, Naïve Bayes Algorithm etc. we can determine about the image is spam or not.

The below definition will be used all over the project.

- Spam: It is the E-mail that contains advertisement and has not been requested.
- Ham: The desired E-mail of our interest is called as Ham.
- False Negative: A spam E-mail message that has not been detected successfully.
- False Positive: A ham E-mail message that has not been detected as spam.

An informative feature subset is selected using Genetic feature search method. The concerned classifier are the ones which are tested on the informative feature subsets that are extracted from three different publicly available dataset.



3:00 p.m.

Fullerton India
to me
12 Nov View details

Why is this message in Spam? It contains content that's typically used in spam messages.

Open In Next Tab

Fullerton India

*Realise your dreams with Fullerton India's Personal Loans starting at 11.99%**

To check your loan eligibility click here.

Fullerton India offers hassle-free, insta Personal Loans that can help you fulfill all your dreams. Hurry! Limited period offer.

Key features:

- Faster and convenient processing
- Minimal documentation
- Disbursal in 72 hours*
- Loans upto 20 Lakhs*

Apply Online Now

FULLERTON INDIA CREDIT COMPANY LIMITED

Registered office address: Megh Towers, 3rd Floor, Old No. 307, New No.165, Poonamallee High Road, Maduravoyal, Chennai 600095, Tamil Nadu.

3:04 p.m.

IRCTC SBI Platinum Card
to me
20 Nov View details

Why is this message in Spam? It's similar to messages that have been detected by our spam filters.

Get Free Railway Tickets with IRCTC SBI Platinum Credit Card!

Exclusive Benefits*:

- 350 bonus Reward Points on spends of Rs.500 or more within 45 days of card issuance
- Upto 10% value back as Reward Points on buying train tickets at irctc.co.in for AC1, AC2 and AC CC categories.
- 1.8% transaction charge waiver every time you book your train tickets at irctc.co.in
- 1 Reward Point for every Rs. 125 spent on non-fuel purchases
- 2.5% fuel surcharge waiver across all petrol pumps in India
- Use your Reward Points to book free railway tickets (1 Reward Point = 1 Rupee)

Travel More. Save More.

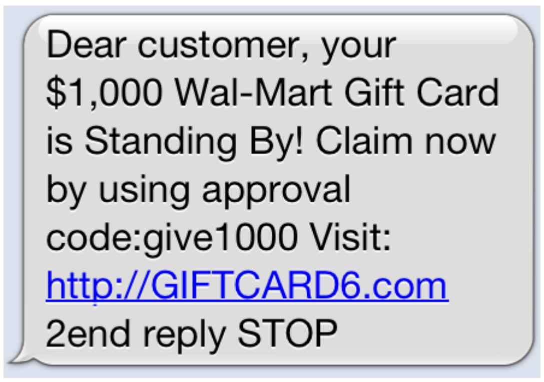
APPLY NOW

SBI Card

* Terms &Conditions apply. For details, log onto sbicard.com
"SBI Card" is a registered logo/trademark of SBICPSL. You may call SBI Card helpline for any additional clarifications or for any other issues with regard to any information contained in this emailer. Please do not reply to this email as it is a computer generated email and reply to this email id is not monitored.

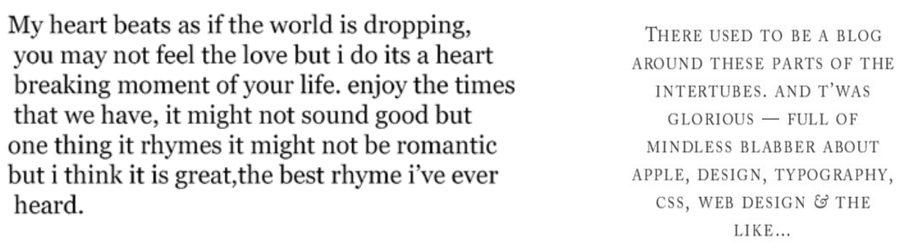
Report Spam | Unsubscribe

Figure 1: Sample Spam Images in E-mails



Dear customer, your $1,000 Wal-Mart Gift Card is Standing By! Claim now by using approval code:give1000 Visit: http://GIFTCARD6.com 2end reply STOP

Figure 2: Sample Spam Image in Message



My heart beats as if the world is dropping, you may not feel the love but i do its a heart breaking moment of your life. enjoy the times that we have, it might not sound good but one thing it rhymes it might not be romantic but i think it is great,the best rhyme i've ever heard.

THERE USED TO BE A BLOG AROUND THESE PARTS OF THE INTERTUBES. AND T'WAS GLORIOUS — FULL OF MINDLESS BLABBER ABOUT APPLE, DESIGN, TYPOGRAPHY, CSS, WEB DESIGN & THE LIKE...

Figure 3: Sample Ham images

II. LITERATURE SURVEY

A lot of literature survey which have been conducted in the area spam classification techniques. Here different methods of image spam filtering are described that has become a critical area of research in recent years. Here the study includes machine learning classifiers such as Naïve Bayes Algorithm and different ways of enhancement of the learning capability of classifiers in a supervised learning environment. Here described a variety of research which have been experimented in literature where many classifiers have been tested on different datasets. The three approaches discussed below are in reference to the research paper, "*Proposed efficient algorithm to filter spam using machine learning techniques*" by Ali Shafigh Aski, Navid Khalilzadeh Sourati, P.P- 145-149, Nov,2016.

II.1 Naïve Bayes Classifier

The Bayesian classifier was introduced by Lewis (1998). This method involves traditional text mining approaches based on content and domain knowledge about the documents. This algorithm is very simple algorithm of classification of spam filtering. This algorithm use the concept of independent events. Consider the events A and B, if we need to use Naïve Bayes Algorithm the events should be independent of each other like one occurrence should not affect the other. In this method of spam filtering we have probability of spam and ham words. After that we train the data and run Naïve Bayes Algorithm in following manner:

$$P(A=a|B=b) = \frac{P(A=a)* P(B=b|A=a)}{P(B=b)}$$

Above,

- P(A=a|B=b) is the probability of A given the probability occurrence of B.
- P(B=b) is the probability of B.
- P(B=a|A=a) is the likelihood which is the probability of B.
- P(A=a) is the probability of A.

Also this Classifier algorithm is extended to n numbers of independent events which can be shown as:

$$P(A=a|B=b)=\pi_i P(A_{i=}a|B_i=b)$$

To get more clarity in the concept of Naïve Bayes consider an example as shown below: We are provided that the Click word comes into 300 out of 1000 spams and also the same word comes in only 10 out of 600 ham mails then the probability of an emails which can contain the word Click is said to be spam is calculated as:

$$P(Click|spam) = \frac{300/1000}{300/1000+10/600} = .9493$$

II.2 C4.5 Decision Tree Classifier

It is clear from the name that the result of the decision tree will be stored in a tree. A C4.5 decision tree is modelled as follows:

A training set is defined for this method which is the collection of base attributes to determine the classes which are related to these attributes. The attributes P can be written as by an vector P=(p1,p2,p3,….pN). Assume that a tuple belongs to a class that is defined earlier which is determine by class label. The set of training is at random selected from the base that is said to be the step of learning. This method of classifier is somewhat efficient.

The structure of the output of the tree is as:

A node represents test which has to done on an adjectives.

An arrow from one node to another in tree show the output.

The leaf node is said to be a class label.

The illustration of decision tree classification is as shown in figure

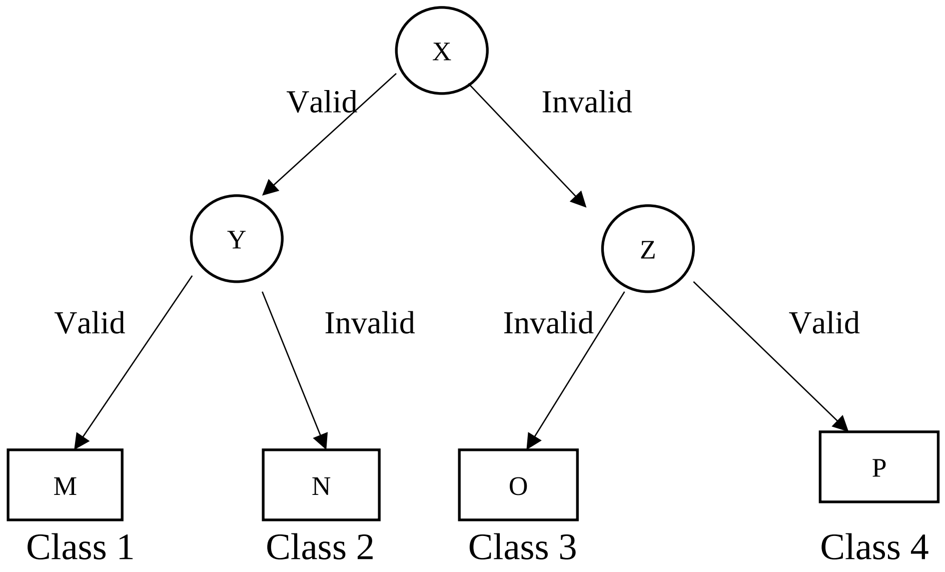


Figure 4: Classification based on decision tree

II.3 Multilayer Perceptron

The multilayer perceptron is shown in diagram. It is a neural network. This model used to deliver the info with the activation of neuron which contain the data labelled on them. The activate of neuron is as calculated

$$P_i=\sigma(\sum_j A_{ij} B_j)$$

Where,

P_i represents the level of activation level of the neuron i;

j is set of data of neuron which we get from the previous layer;

A_{ij} is said to be the weight of the connection between the different neuron i and j;

B_j used to represent the result we get from the neuron j;

σ(x) used to represent the transfer function.

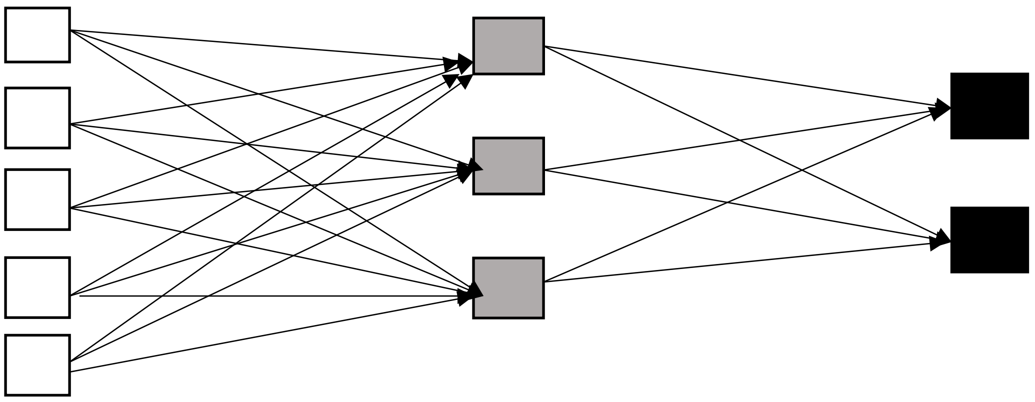


Figure 5: Multilayer Neuron Network

The algorithm is not linear between the inputs and outputs. This is made to happen by connecting the neuron of a node to its previous and its next layers as shown in the diagram above. The output obtained are multiplied by the coefficients of weights. After doing this these are insert into the nonlinear function of activation which is written above in this algorithm. The one thing that is important for this algorithm is that it usually solve the problem when a fixed model or knowledge is not provided on input and their relation with outputs.

III. CONCEPTS AND ALGORTHM

III.1 Naïve Bayes Algorithm

It is basically a classifier that assumes that occurrence of a particular event is independent of the occurrence of other event. It is based upon the assumption of independence among the predictors. For example, a fruit which is red, round and 3 inches in diameter may be considered apple. Even if these feature depend upon existence of other feature or on each other, all these properties independently contribute to the probability that this fruit is an apple and therefore it is called as ‘Naïve’. Naïve Bayes is mainly used for very large dataset and it is easy to build model.

Algorithm for Image Based Classifier

1. Extract text from image using OCR (optical character recognition).
2. Run this text message on the Naive-Bayes Classifier for test.

Naïve Bayes for Spam classifier

Notations

Word$_i$= w_i

Spam= s

Ham= h

Probability= P

Mesasage= m

Proposed Algorithm

$m = <w_1, w_2, w_3, \dots , w_n>$

$P(s \mid m) = P(w_1 \mid s) * P(w_2 \mid s) * \dots * P(w_n \mid s) * P(s) / P(m)$

$P(h \mid m) = P(w_1 \mid h) * P(w_2 \mid h) * \dots * P(w_n \mid h) * P(h) / P(m)$

Taking log of the above equations

$\log_{10} (P(s \mid m)) = \log_{10} (P(w_1 \mid s)) + \log_{10} (P(w_2 \mid s)) + \dots +$
$\log_{10} (P(w_n \mid s)) + \log_{10} (P(s)) - \log_{10} (P(m))$

$\log_{10} (P(h \mid m)) = \log_{10} (P(w_1 \mid h)) + \log_{10} (P(w_2 \mid h)) + \dots +$
$\log_{10} (P(w_n \mid h)) + \log_{10} (P(h)) - \log_{10} (P(m))$

if $\log_{10}(P(s \mid m)) < \log_{10}(P(h \mid m))$

then spam

otherwise ham

Generating Training Data required by the Classifier

1. First of all, a data set of 1000 spam emails and 900 ham emails is prepared.
2. All the entities, phrases that do not have any meaning like Phone No., E-mails and website links are removed.
3. All the special characters are removed except ‘$’.
4. The words collected so far are stemmed and converted to root words (like *studied* converted to *study*) as they all have the same meaning.
5. A program is implemented which maintains two hash map (word string against its number of occurrence in spam and ham)
6. The probability of each word obtained so far is calculated by dividing its number of occurrences by total number of words. This is stored in the hash map with the key as word and probability as its value.
7. The classifier program uses these hash maps.
8. To optimize the searching time of map, Trie data structure can be used.

Classifier Algorithm

1. Two consecutive words are taken from the message at a time to make a new word or sentence.
2. The three words that we have right now the two independent word and one new word should be looked up in the hash map for their probability.
3. The log of all these probabilities is taken and added.
4. If the log (P (spam | message)) is greater than or equal to log (P (ham | message)), then the message is ham otherwise spam.

IV. DATA FLOW DIAGRAMS:

IV.1 Level 0 DFD

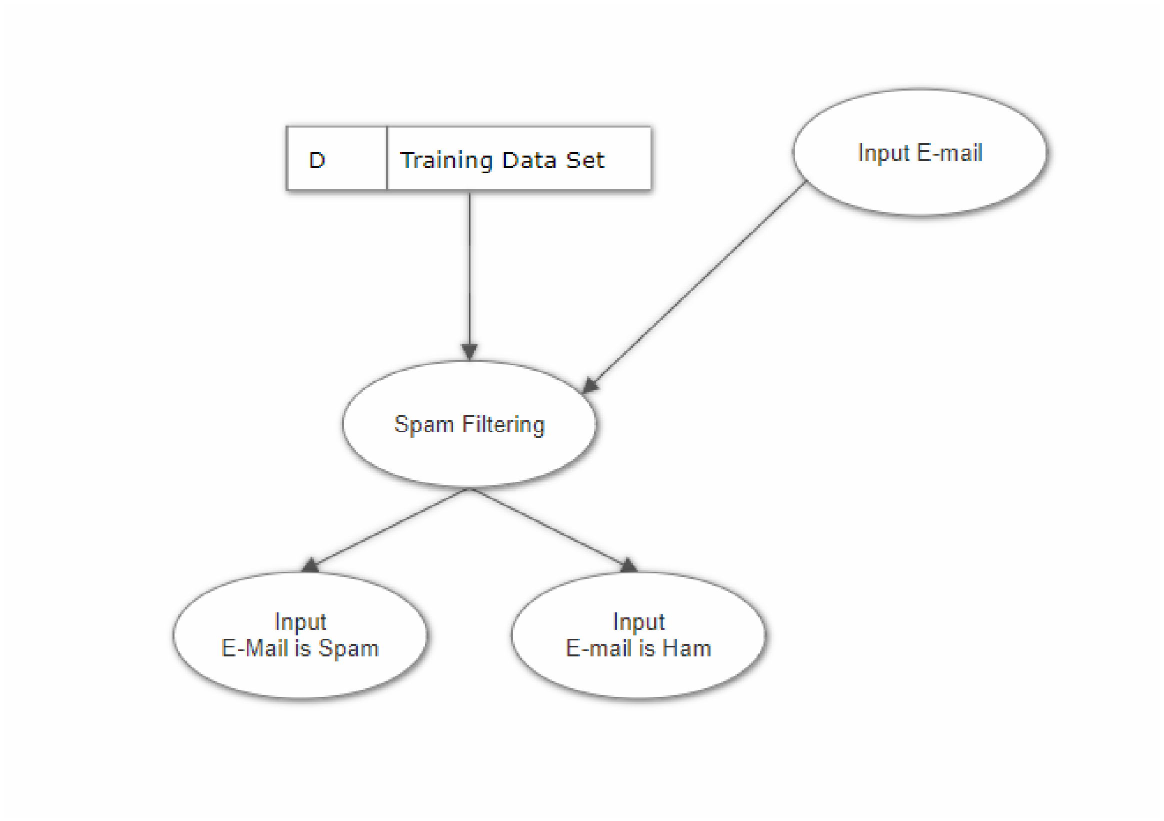
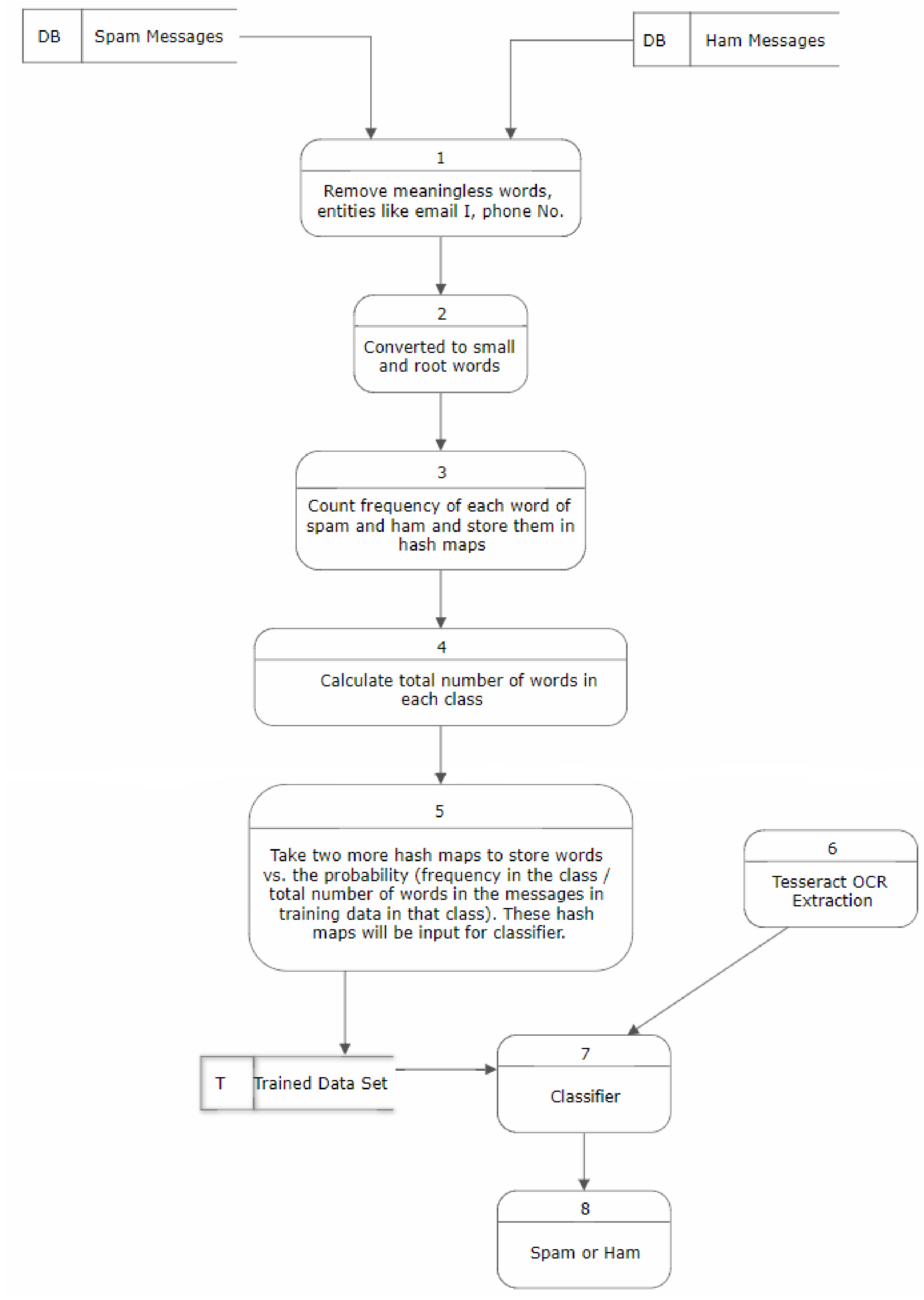


Figure 6: Level 0 DFD

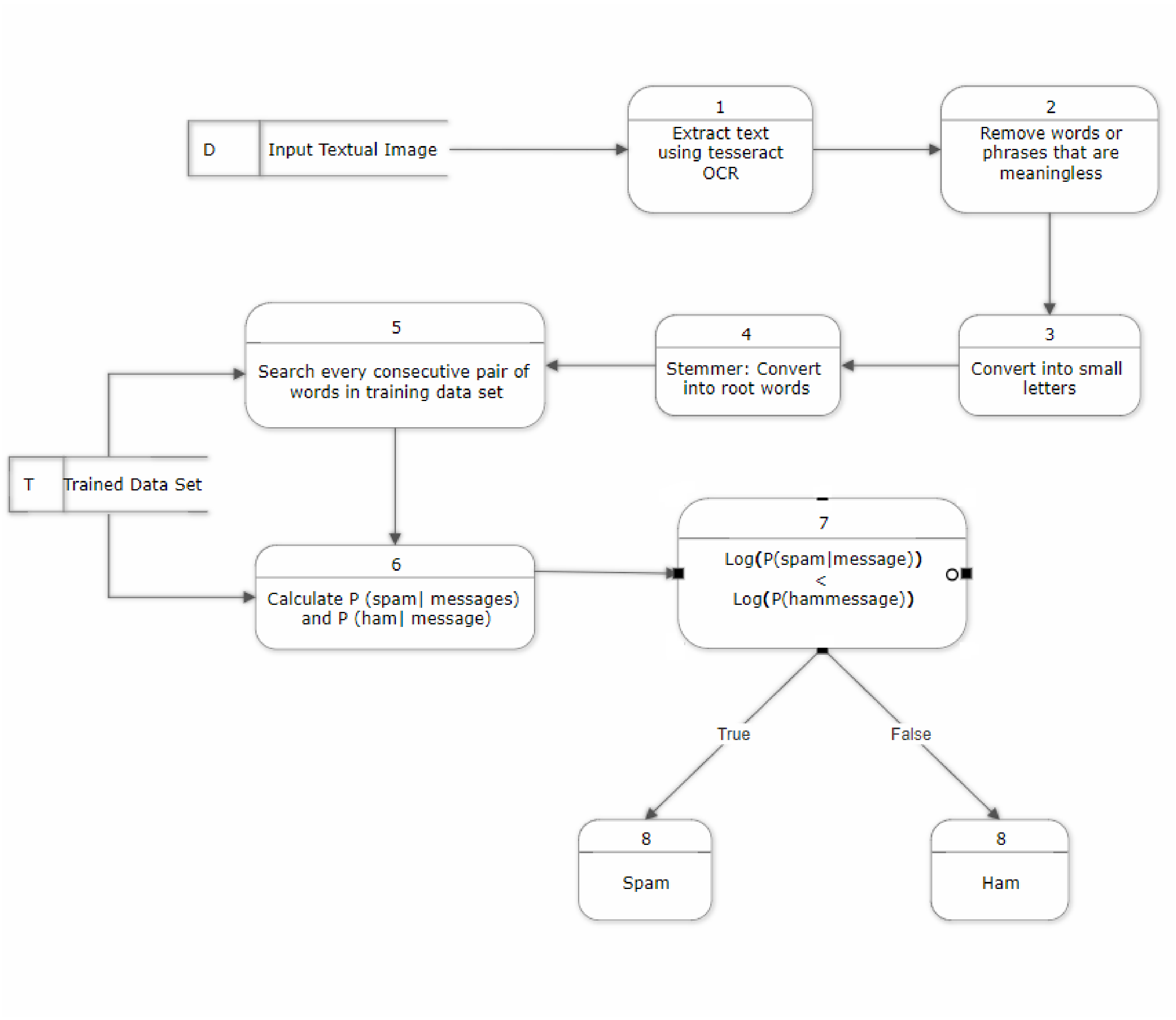
IV.2 Level 1 DFD



DB
Spam Messages
DB
Ham Messages
1
Remove meaningless words, entities like email I, phone No.
2
Converted to small and root words
3
Count frequency of each word of spam and ham and store them in hash maps
4
Calculate total number of words in each class
5
Take two more hash maps to store words vs. the probability (frequency in the class / total number of words in the messages in training data in that class). These hash maps will be input for classifier.
6
Tesseract OCR Extraction
T
Trained Data Set
7
Classifier
8
Spam or Ham

Figure 7: Level 1 DFD

IV.3 Level 2 DFD

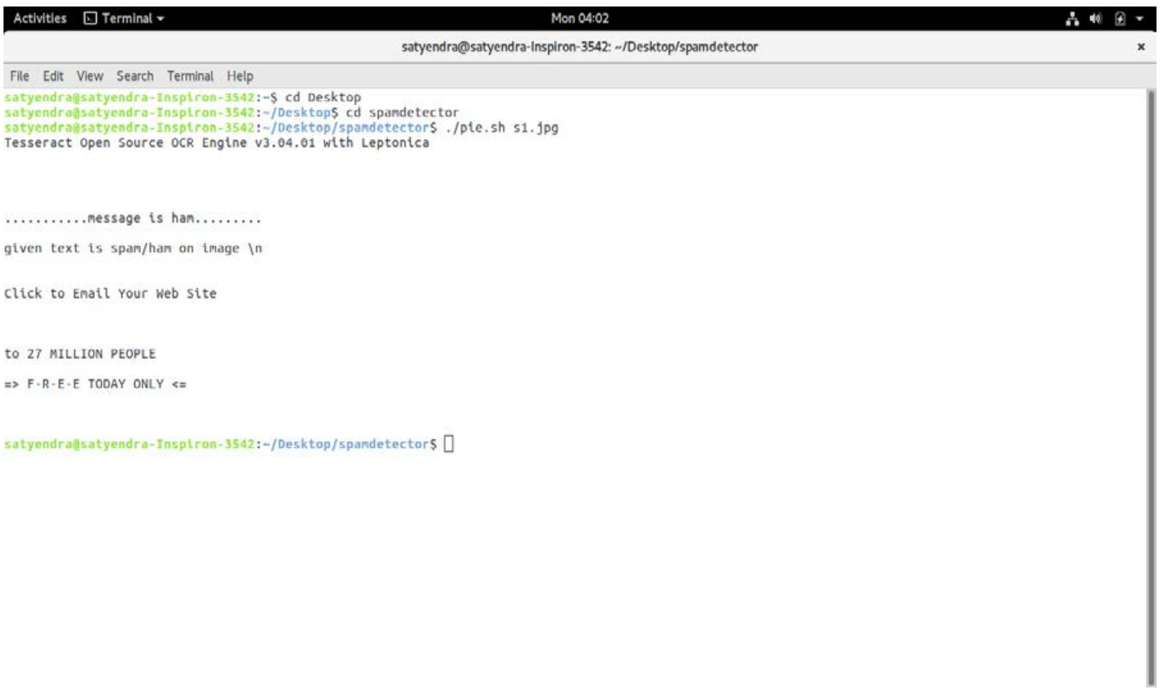


D
Input Textual Image
1
Extract text using tesseract OCR
2
Remove words or phrases that are meaningless
3
Convert into small letters
4
Stemmer: Convert into root words
5
Search every consecutive pair of words in training data set
T
Trained Data Set
6
Calculate P (spam| messages) and P (ham| message)
7
Log(P(spam|message)) < Log(P(hammessage))
True
False
8
Spam
8
Ham

Figure 8: Level 2 DFD

V. RESULTS AND OBSERVATIONS

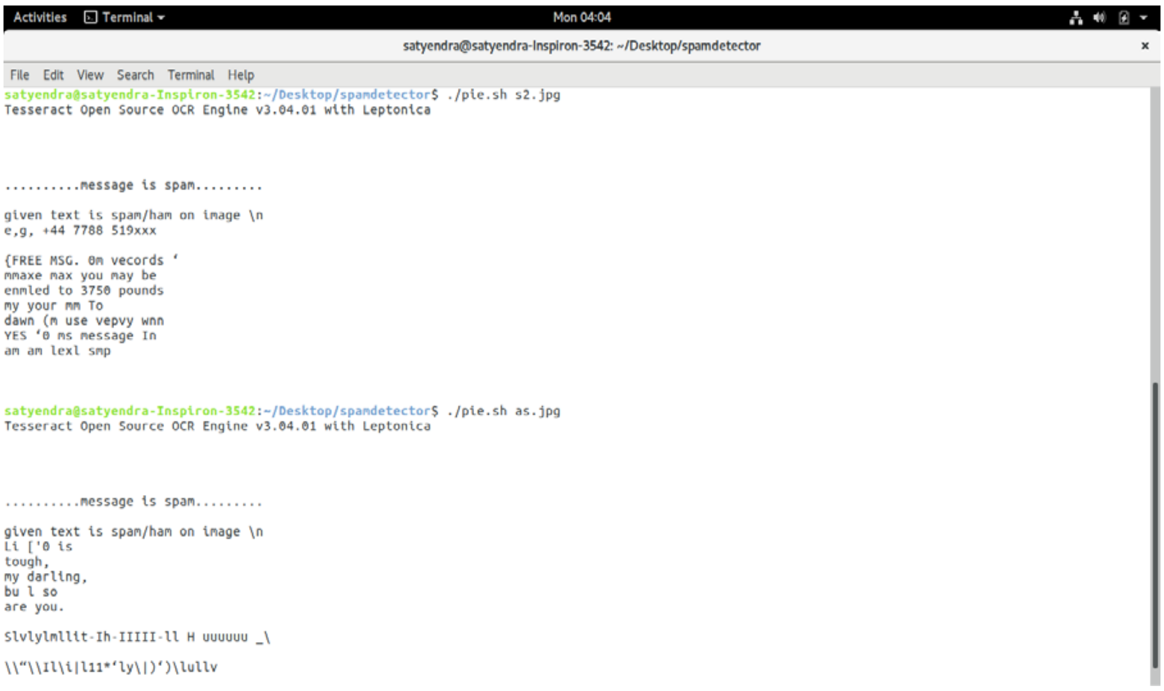
The classifier trained on data works fine. The classifier trained on data determines whether the image is spam or ham. Following are some screenshots of the results:


Activities
Terminal
Mon 04:02
satyendra@satyendra-Inspiron-3542: ~/Desktop/spamdetector
File Edit View Search Terminal Help

```
satyendra@satyendra-Inspiron-3542:~$ cd Desktop
satyendra@satyendra-Inspiron-3542:~/Desktop$ cd spamdetector
satyendra@satyendra-Inspiron-3542:~/Desktop/spamdetector$ ./pie.sh s1.jpg
Tesseract Open Source OCR Engine v3.04.01 with Leptonica



...........message is ham.........

given text is spam/ham on image \n

Click to Email Your Web Site


to 27 MILLION PEOPLE

=> F-R-E-E TODAY ONLY <=


satyendra@satyendra-Inspiron-3542:~/Desktop/spamdetector$
```

Figure 9: Screenshot of classifier running on a sample ham image

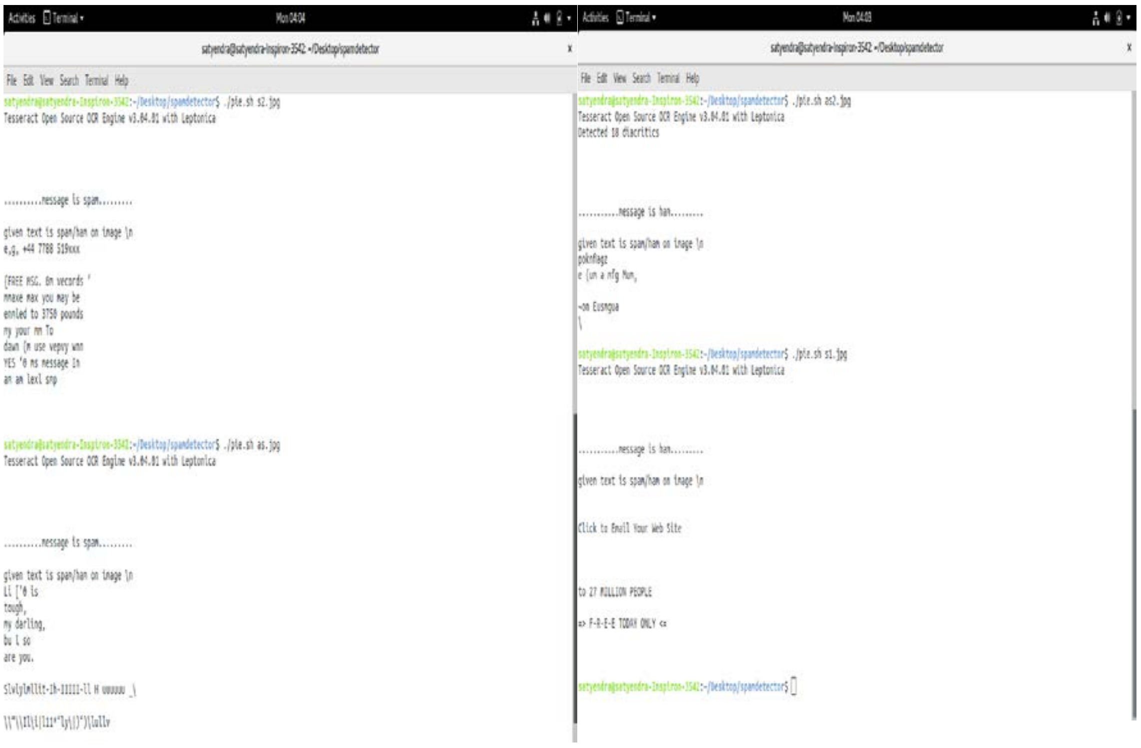
The above Figure contains the input image, text extracted from the input image by tesseract OCR API and the terminal running the classifier, which detected the input image as ‘Ham’.


Activities
Terminal
Mon 04:04
satyendra@satyendra-Inspiron-3542: ~/Desktop/spamdetector
File Edit View Search Terminal Help

```
satyendra@satyendra-Inspiron-3542:~/Desktop/spamdetector$ ./pie.sh s2.jpg
Tesseract Open Source OCR Engine v3.04.01 with Leptonica



..........message is spam.........

given text is spam/ham on image \n
e,g, +44 7788 519xxx

{FREE MSG. 0m vecords ‘
mmaxe max you may be
enmled to 3750 pounds
my your mm To
dawn (m use vepvy wnn
YES ‘0 ms message In
am am lexl smp


satyendra@satyendra-Inspiron-3542:~/Desktop/spamdetector$ ./pie.sh as.jpg
Tesseract Open Source OCR Engine v3.04.01 with Leptonica



..........message is spam.........

given text is spam/ham on image \n
Li ['0 is
tough,
my darling,
bu l so
are you.

Slvlylmllit-Ih-IIIII-ll H uuuuuu _\

\\“\\Il\i|l11*‘ly\|)‘)\lullv
```

Figure 10: Screenshot of classifier running on a sample spam image

The above Figure 7 contains the input image, text extracted from the input image by tesseract OCR API and the terminal running the classifier, which detected the input image as ‘Spam’.



Activities Terminal ▾
Mon 04:04
satyendra@satyendra-Inspiron-3542: ~/Desktop/spamdetector
File Edit View Search Terminal Help

```
satyendra@satyendra-Inspiron-3542:~/Desktop/spamdetector$ ./pie.sh s2.jpg
Tesseract Open Source OCR Engine v3.04.01 with Leptonica



..........message is spam.........

given text is spam/ham on image \n
e,g, +44 7788 519xxx

[FREE MSG. 0n vecords '
mnaxe max you may be
ennled to 3750 pounds
my your mn To
dawn [m use vepyy wnn
YES '0 ms message In
an am lexl snp



satyendra@satyendra-Inspiron-3542:~/Desktop/spamdetector$ ./pie.sh as.jpg
Tesseract Open Source OCR Engine v3.04.01 with Leptonica



..........message is spam.........

given text is spam/ham on image \n
Ll ['0 is
tough,
my darling,
bu l so
are you.

SlvlylmlIlt-1h-IIIII-ll H uuuuuu _\

\\"\\Il\l(111*"ly\|)")\lullv
```

Activities Terminal ▾
Mon 04:03
satyendra@satyendra-Inspiron-3542: ~/Desktop/spamdetector
File Edit View Search Terminal Help

```
satyendra@satyendra-Inspiron-3542:~/Desktop/spamdetector$ ./pie.sh as2.jpg
Tesseract Open Source OCR Engine v3.04.01 with Leptonica
Detected 18 diacritics



...........message is ham.........

given text is spam/ham on image \n
poknfagz
e {un a nfg Mum,

~on Eusngua
\

satyendra@satyendra-Inspiron-3542:~/Desktop/spamdetector$ ./pie.sh s1.jpg
Tesseract Open Source OCR Engine v3.04.01 with Leptonica



...........message is ham.........

given text is spam/ham on image \n


Click to Email Your Web Site



to 27 MILLION PEOPLE

=> F-R-E-E TODAY ONLY <=



satyendra@satyendra-Inspiron-3542:~/Desktop/spamdetector$ 
```

Figure 11: Classifier running on test data

The above screenshot tests various images if they are spam or ham based on the probabilities calculated by the classifier.

VI. CONCLUSION AND FUTURE PLAN

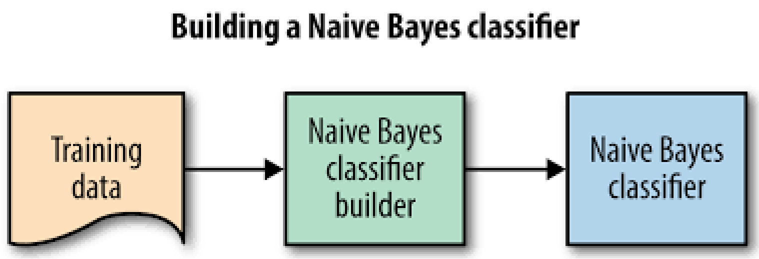
Conclusion

The project makes it clear the general idea of Image spam and how it works. It is produces difficulty in various aspects and erodes the network resources considerably; which is why this project has introduced a feature extraction scheme that rely on low-level features.

This project also sees how the spam detection works and the steps done to filter and eventually block it. So far, it is able to extract both the text and image features from a given image and give it into classifier. The classifier then decides whether the given input image is a spam or ham. We chose Naïve Bayes Algorithm as the image spam detection is significantly more accurate as compared with other similar algorithms.

The Naïve Bayes Classifier is used to identify whether the input image is an image spam or not. Although, the extraction scheme is known to be efficient in this algorithm, the same can't be said for accuracy. This algorithm faces difficulty in accuracy because of corrupted and multi frame images which is why it won't give an efficient result. On the other hand, the false positive and false negative factors are also efficient. If a spam image is considered a ham, then it's called false positive. And if the ham image is considered as spam, it's called false negative.

This study describes three matching algorithms to filter spam from valid mails with low error rates and high efficiency using Naive Bayes algorithm. Several other techniques are also there like C4.5 decision tree, multilayer perceptron, all of which are used for training data whether in the form of spam or valid emails.


Building a Naive Bayes classifier
Training data
Naive Bayes classifier builder
Naive Bayes classifier

Future Plan

There is a lot of scope and room for improvement that can be made to this project. In future, additional features can be added and more optimization techniques can be used that can speed-up the detection rate. By doing so, the time complexity will be significantly reduced. The project should be able to work well with multi-frame images. The application developed should be flexible and the changes can be made wherever required.

To improve the accuracy and consistency rate, more algorithms like Multi-Layer Perceptron and Decision Tree can be implemented which will yield a better and far more accurate result. The drawbacks and limitation of the current system will be lifted and new opportunities will be opened. By adding more algorithms, the system can cross-check the result with each other and ultimately, provide the correct proven result. Machine learning can be added which will prove to be a great boon over time. The data that the system will gain after running various images over time will only help in improving the overall accuracy. The other big change that can increase the speed of detection and classification is by the use of tri-data structure. This will drastically improve the search time and complexity of the system.

Most of the objectives and features of the project have been dealt within the estimated schedules and quality features. The project was verified with both valid and invalid data in each manner. The applications also runs with an insight into the necessary modifications that may be required in the near future. Hence, the system can be maintained successfully.

VII. REFERENCES

[1] Ali Shafigh Aski Navid Khalilzahed Sourati, “Proposed efficient algorithm to filter spam using machine learning techniques”, P.P. 145-149, Nov., 2016

[2] http://ieeexplore.ieee.org/document/4376991/authors

[3] Battista Biggio, et. al. “A survey and experimental evaluation of image spam filtering techniques”, P.P. 1436-1446, July, 2011

[4] YanGao, MingYang, Xiaonan Zhao, Bryan Pardo, YingWu, ThrasyvoulosN.Pappas, Alok Choudhary, EECS Dept., Northwestern Univ. 2145 SheridanRd. Evanston, IL 60208 – Image Spam Hunter, 2008

[5] https://spamassassin.apache.org/publiccorpus

APPENDIX

COMPLETE CONTRIBUTARY SOURCE CODE

Test_english.cpp

Used to convert the words into their root.

```cpp
#include <iostream>
#include <string>
#include "stemming/english_stem.h"
using namespace std;
int main()
{
   string str;
   while ( cin >> str )
   {
      wstring word(str.begin(), str.end());
      stemming::english_stem<> eng;
      eng(word);
      wcout << word.c_str() << "\n";
   }
   return 0;
}
```

Converter.cpp

Used to convert the text into their lower-case and eliminate the meaningless phrases.

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
   ifstream infile;
   infile.open("output_stemmer.txt");
   ofstream outfile;
   outfile.open("newfile.txt");
   string str;
   while(cin>>str != NULL)
   {
      int i = 0;
      while(i < str.length())
      {
         if(str[i] <= 'Z' && str[i] >= 'A')
         {
            str[i] += 32;
            cout<<str[i];
         }
         else if(str[i] == '$')
            cout<<" "<<str[i]<<" ";
         else if(str[i] <= 'z' && str[i] >= 'a')
            cout<<str[i];
         i++;
      }
      cout<<" ";
   }
   return 0;
}
```

Training data

Used to train the dataset obtained from spam and ham e-mails

```
#include<bits/stdc++.h>
using namespace std;
#define LL long long
#define S 1000005
#define W while
#define M 1000000007
#define str string
#define pi 3.14159265
LL mini=INT_MAX;
LL maxi=INT_MIN;
int primes[10000001];
using namespace std;
map<string, int> m;
map<string, double> m_prob;
int main()
{
   LL total_words=0,total_ham=0,total_spam=0;
   ifstream  afile;
   ofstream outfile;
       outfile.open("train_prob_spam");
   afile.open("spam_stemmed.txt", ios::out | ios::in );
   str s1,s2;
   afile>>s1;
   while(afile>>s2)
   {
      m[s1]++;
      m[s1+s2]++;
      s1=s2;
   }
   map<string ,int> ::iterator it=m.begin();
   for(;it!=m.end();it++)
```

```
{
   total_words+=it->second;
}
total_spam=total_words;
for(it=m.begin();it!=m.end();it++)
{
   m_prob[it->first]=(double)it->second/total_words;
}

map<string, double> :: iterator it1;

for(it1=m_prob.begin();it1!=m_prob.end();it1++)
{
   outfile<<it1->first<<" "<<it1->second<<endl;
}
outfile.close();
afile.close();
m.clear();
m_prob.clear();

outfile.open("train_prob_ham");
afile.open("ham_stemmed.txt", ios::out | ios::in );
afile>>s1;
while(afile>>s2)
{
   m[s1]++;
   m[s1+s2]++;
   s1=s2;
}
it=m.begin();
total_words=0;
for(;it!=m.end();it++)
{
   total_words+=it->second;
```

```
    }
    total_ham=total_words;
    for(it=m.begin();it!=m.end();it++)
    {
       m_prob[it->first]=(double)it->second/total_words;
    }


    for(it1=m_prob.begin();it1!=m_prob.end();it1++)
    {
       outfile<<it1->first<<" "<<it1->second<<endl;
    }
    afile.close();
    outfile.close();
    return 0;
}
```

Classifier

Used to classify the e-mail image text into spam or ham.

```
#include<bits/stdc++.h>
using namespace std;
#define LL long long
#define S 1000005
#define W while
#define M 1000000007
#define str string
#define pi 3.14159265
LL mini=INT_MAX;
LL maxi=INT_MIN;
using namespace std;
map<string, int> m;
map<str,double> spam,ham;
vector<str> message;
int main()
{
   ifstream infile,spamfile,hamfile;
   infile.open("newfile.txt");
   spamfile.open("train_prob_spam");
   hamfile.open("train_prob_ham");
   str s;
   double d;
   while(spamfile>>s)
   {
      spamfile>>d;
      spam[s]=d;
   }
   while(hamfile>>s)
   {
      hamfile>>d;
      ham[s]=d;
```

```
}
string s1;
W(cin>>s1)
{
   message.push_back(s1);
}
double hp=0,sp=0;
map<str,double> ::iterator it;

for(int i=0;i<message.size()-1;i++)
{
   string word1, word2, word3;
   word1 = message[i], word2 = message[i + 1];
   word3 = word1 + word2;

   it = spam.find(word1);
   if ( it != spam.end() )
   {
      sp += -1*log10(it -> second);
   }

   it = spam.find(word3);
   if ( it != spam.end() )
   {
      sp += -1*log10(it -> second);
   }

   it = ham.find(word1);
   if ( it != ham.end() )
   {
      hp += -1*log10(it -> second);
   }

   it = ham.find(word3);
```

```
      if ( it != ham.end() )
      {
         hp += -1*log10(it -> second);
      }
   }
    if(spam.find(message[message.size()-1])!=spam.end())
      sp+=-1*log10(spam[message[message.size()-1]]);
   if(ham.find(message[message.size()-1])!=ham.end())
      hp+=-1*log10(ham[message[message.size()-1]]);

    sp+=
1*log10((double)spam.size()/(spam.size()+ham.size()));
    hp+=-
1*log10((double)ham.size()/(ham.size()+spam.size()));

   if(sp>hp)
      cout<<"\n\n\n\n..........message is
spam.........\n\n";
   else cout<<"\n\n\n\n...........message is
ham.........\n\n";

   spamfile.close();
   hamfile.close();
   infile.close();
}
```

pie.sh

The script that automates the whole project.

```
tesseract "$1" out
./converter <out.txt > a1
./stemmer <a1 > b1
./classifier <b1
echo "given text is spam/ham on image \n"
cat out.txt
```