SENTIMENT ANALYSIS ON VISUAL DATA

Thesis submitted in partial fulfilment of requirements

for the award of the degree of

Master of Technology
in
Computer Science and Engineering

under the esteemed guidance of
Prof. Rajni Jindal
(Head of Department
– Computer Science and Engineering)
Delhi Technological University

Submitted By-
Garima
(Roll No. - 2K16/CSE/06)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

SESSION: 2016-2018

DECLARATION

I hereby declare that the thesis work entitled "**Sentiment Analysis on Visual Data**" which is being submitted to Delhi Technological University, in partial fulfilment of requirements for the award of degree of Master of Technology (Computer Science & Engineering) is a bonafide report of thesis carried out by me. The material contained in the report has not been submitted to any university or institution for the award of any degree.

Garima
2K16/CSE/06

CERTIFICATE

This is to certify that report entitled Garima (2K16/CSE/06) has completed the thesis titled "**Sentiment Analysis on Visual Data**" under my supervision in partial fulfilment of the MASTER OF TECHNOLOGY degree in Computer Science & Engineering at DELHI TECHNOLOGICAL UNIVERSITY.

Supervisor

Prof. Rajni Jindal
Head of Department
Department of Computer Science and Engineering
Delhi Technological University
Delhi -110042

ACKNOWLEDGEMENT

I am very thankful to Prof. Rajni Jindal (Head of Department, Computer Science Eng. Dept.) and all the faculty members of the Computer Science Engineering Dept. of DTU. They all provided immense support and guidance for the completion of the project undertaken by me.

I would also like to express my gratitude to the university for providing the laboratories, infrastructure, testing facilities and environment which allowed me to work without any obstructions.

I would also like to appreciate the support provided by our lab assistants, seniors and peer group who aided me with all the knowledge they had regarding various topics.

Garima

M. Tech. in Computer Science & Engineering

Roll No. 2K16/CSE/06

ABSTRACT

User generated content on the internet is widely diverse as well as massive in terms of amount of data being uploaded each day. One way to extract information out of that huge pile of data is to perform sentiment analysis on it. A great deal of time and extensive research has been dedicated to textual data sentiment analysis in the past decades. With the advent of social media sites like Flickr and Instagram that majorly rely on uploading and sharing visual data, there is a need to focus on sentiment analysis that also incorporates visual attributes. Analysis of images uploaded on these sites along with labels and tags can give us a hint about the mood of the users related to a particular topic or event that can further our goals in collecting an overall public sentiment poll. In a way, visual sentiment analysis can work alongside textual sentiment analysis to gain a better insight.

In order to achieve that goal, this report uses an approach similar to Visual Sentiment Ontology (VSO). A large scale of visual data along with related tags was manually extracted from Flickr to create a library of Adjective-Noun Pairs (or ANPs). Image-content specific features were also retrieved from the images which convey the visual properties of the image like texture, shape etc. The Adjective-Noun Pairs and the image features help to gauge the emotion in the images and divide them in distinct eight sentiment categories. These eight sentiment categories correspond to the famous 'Plutchik's Wheel of Emotions' that has been a huge success in psychological studies. Limited work has been done in this area so far, with little efficiency. So, this paper applies supervised machine learning algorithms to improve the performance of the current research and also introduces a new avenue which incorporates a mix of ANP and image attributes to solve the sentiment prediction problem. Additional effort was put into generating association rules between the attributes and applying specific visual filters to enhance the images and retrieve image attributes.

Weka (Waikato Environment for Knowledge Analysis) tool is used for classification of the images. Weka is software that is a comprehensive collection of machine learning algorithms written in Java, and has been developed at the University Of Waikato, New Zealand. It incorporates various tools for pre-processing of data, classification algorithms, clustering

algorithms, association rules, and easy to perceive visualization. These algorithms are used to learn from examples or "training set", and classify new sets into categories analyzed. Weka is open source software issued under the GNU General Public License [19].

Java is used to pre-process the data set and enhance images, while MATLAB is also partially used to visualize the results in a user friendly manner.

A comparison analysis was conducted between the approach used in this research and the latest studies done in this area. The comparison was tested against various classification algorithms and different kind of variations of chosen attributes. The result clearly indicated that our approach works significantly better than the recent studies.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
Introduction

In today's world, a major part of information exchange happens on the Internet. This information is passed in the form of images, comments, tags, videos etc. To devise innovative and efficient methods to gather meaningful knowledge out of that abundant amount of data is the need of the hour. Sentiment analysis is one such method to collect useful information from the user generated content on the Internet and gauge the basic human emotions behind it in a systematic way. It helps in understanding the human decision making and getting a broader perspective of the public on a particular subject.

Till now, most of the research work has focussed mainly on textual sentiment analysis but as we are evolving as a global Internet community, we are constantly generating heavy-scale visual content over the network. Sites like Flickr, Instagram, Snapchat and Facebook are all the rage among the millennial generation. To give an estimate, Flickr recorded over 75 million registered photographers as by 2017, who upload around 25 million photos on a very high traffic day [22]. Thus, there is an obvious need to include visual attributes as well to sentiment analysis.

We propose to discover and detect visual attributes from the images as well as the tags associated with them and classify the image-data set with their help. The tags are converted into Adjective-Noun Pairs to bring forward the emotions depicted in the image. The images are then classified into eight categories corresponding to the eight emotions of the famous 'Plutchik's Wheel of Emotions'. Supervised machine learning algorithms and specific image filters, based on color distribution, edges present in the images and other similar properties, provide us with better results than the current research in this area.

1.1 RESEARCH CHALLENGES

We explore the subsequent research challenges that come across in this study:

- *RC1:* Human emotions are pretty abstract already. Different cultures may stimulate different emotions with respect to a single image. That clarified, we must acknowledge that semantic concepts of an image can be very difficult to obtain from the content concepts of an image. In other words, there is a gap between the meaning or output emotion (or high level concept) invoked by an image and the objects contained in the image like 'fire' or 'cat' (low level concepts).

  This challenge is overcome by introducing Adjective-Noun Pairs derived from the tags and comments to bridge the gap between low level and high level concepts.

- *RC2:* It is troublesome to compress wide variety of human emotions into just two or three basic categories such as positive, negative or neutral. Even if we succeed in assigning the emotions as positive or negative, it won't do justice to their essence. For example, we can't simply put the emotion 'anger' into the 'negative' category, as that anger stimulated by an image can be rightful or 'positive' if it depicts a mob of angry citizens demonstrating against some human right violation. To overcome this challenge, we must consider sentiment as a spectrum or at least a multivariate function of human emotions than a canonical binary function. We have used the 'Plutchik's Wheel of Emotions' which consists of eight primary emotions which serve as the foundation elements in extensive psychological research.

- *RC3:* The images available on the internet are diverse in terms of color components, dimensionality, file format etc and thus can render the whole process useless if they are unable to be read by the algorithm. Even if we choose one single file format throughout the dataset, such as 'JPEG', it can still be unreadable by the machine if the number of source raster bands and source color space components do not match.

  The main goal here is to generate similar type of images with features that are readable by the algorithm. Sometimes the color profile of the images can be different which can cause problems later. This can be solved by converting all the images into a single color profile set of a particular file format. This is achieved by running all the images through a Java code that correct its raster bands and standardize the color components.

1.2 MOTIVATION OF STUDY

As we know, research in this area is increasing at a high rate and attracting both the current industry. All the big organizations want to get an honest opinion about their products and services, and Internet is an effective resource of public feedback. People upload the popular opinion in form of images, videos, comments and tags. The majority of research work has relied on textual data mining only. Also, all of the methodologies yet worked upon exhibit limitations in terms of accuracy, robustness and inclusiveness of various attributes. This renders them ineffectual in real world applications.

The aim of including visual attributes is to increase the accuracy and effectiveness of sentiment analysis. It can have many applications in business analytics, tracking product reviews and getting insights about electoral polls etc. Further, there are several forces that push us forward to work in this area.

First, with the advent of social media, we have been provided with a rich repository of user generated image data set like never before. This offers us an opportunity to mine the knowledge pattern of visual sentiments.

Second, human cognition naturally inclines us to study and conceive our perception of visual images. We are psychologically wired to capture emotions in any visual scenario.

Third, today's computer vision technologies are witnessing a never seen before advancement which can help boost the research work in this area.

In this paper, we focus on the current research work and introduce different algorithms and filters to improve their efficiency.

1.3 ORGANIZATION OF THESIS

The thesis is organized in various chapters as follows: Chapter 2 gives an overview of the related work of the study. It discusses the various research works that have been done in this area and how all that work helped in evolution of our study. Chapter 3 summarizes research methodologies used in this thesis including the practicality of Plutchik's Wheel of Emotions and the description of all the Adjective Noun Pairs and other visual attributes used. Chapter 4 describes the proposed solution by describing the dataset, algorithms and image filters used and solution representation. Chapter 5 states result of the study as well as the comparative analysis with current research work and performance evaluation of our study. And at last, chapter 6 summarizes the research work under conclusion and suggests some future work.

CHAPTER 2

Literature Review

This chapter gives an overview of the research work done with relation to our thesis and is further sub divided into two sections i.e. current approaches towards visual sentiment analysis and current visual datasets available.

2.1 CURRENT APPROACHES TOWARDS VISUAL SENTIMENT ANALYSIS

The research done so far can be divided into three categories on the basis of the method of approach chosen to tackle the problem: mid level sentiment ontology, sentiment prediction using deep neural networks and multimodal sentiment prediction.

2.1.1 Mid level sentiment ontology

Yuen et al. proposed a Sentribute approach to combat the problem of sentiment analysis [12]. As mid level features, this approach uses scene related attributes of an image. Four types of mid level features used are material, action, surface property and spatial envelope. Sentribute consists of 102 mid level features which are easily understandable and completely prepared to dispose. To improve accuracy, authors combined the Sentribute approach with facial emotion detection. But it proved to be less rigorous in comparative analysis and therefore less used practically.

Chen et al. proposed a visual sentiment analysis based on objects [2]. It is basically a hierarchical system which uses object localization and sentiment related concept modelling to decompose the large problem. It classifies in an object specific way. Object detectors are used to extract the object in a given image. Later, ANP classifiers are applied on the extracted nouns.

Cao and Ji used Visual Sentiment Topic Model (VSTM) approach [2]. It uses multi image information in a particular topic to construct Visual Sentiment Topic (VST), which is the visual sentiment ontology information for that specific topic. Advantage of VSTM is that it deals with different sentiment expressions of a single topic.

We have mainly focussed on the work of Borth and Ji [1]. Their work uses Adjective Noun Pairs extracted from the annotations of the image data set to build a Visual Sentiment Ontology (VSO). In their original research, they created a SentiBank from VSO. SentiBank is a library of trained concept detectors which provides a mid level feature representation and helps in predicting the sentiment of an image.

2.1.2 Sentiment prediction using deep neural networks

You et al. acknowledged the domain transfer problem and introduced a Progressive Convoluted Neural Network (PCNN) [5]. It transfers the knowledge to other domains using a simple fine tuning approach. Advantage of this algorithm is its generalizability to encapsulate other different domains.

Narihira et al. proposed FactNet [15], a factorized model for ANPs which is based on the constitution of two deep neural networks. Because of factorization, the FactNet provides regularization among different ANPs and even introduces and classifies new ANPs on its own.

2.1.3 Multimodal Sentiment Prediction

To increase the sentiment prediction veracity and performance, a Multimodal Correlation Model (MCM) was proposed by Li et al. to combine multimodal information with sentiment analysis [2]. Multimodal features in forms of words, ANPs and symbols are extracted and Markov Random Field is used to find correlation between modalities.

To improve the aforementioned multimodal method, another research was later done by Chen at al. [2]. They proposed a multimodal hypergraph learning model. This considers the multimodality as well as the independence of modality too. Independence of each modality is learned through hypergraph learning model.

A new scope in this field is unsupervised sentiment analysis for social images. Wang et al. has recently proposed an unsupervised sentiment analysis model (USEA). It reduces semantic gap between low level and high level features [14].

2.2 CURRENT VISUAL DATASETS AVAILABLE

There have been only two effective visual datasets till now. One is visual sentiment ontology dataset and other is a GIF ontology dataset.

2.2.1 Visual Sentiment Ontology Dataset

VSO Dataset was prepared by Borth and Ji for their VSO framework. This dataset extracted 603 images from Flickr, covering 21 different topics. Total 3244 Adjective Noun Pairs form the VSO and the SentiBank consists of 1200 trained visual concept detectors. It has become a benchmark for the visual data sentiment analysis.

But since it was not available easily on the Internet, we constructed a similar dataset manually.

2.2.2 GIF Ontology Dataset

This dataset is used for sentiment analysis on GIF data. It is a unique implementation of GIF sentiment ontology (GSO) and it uses SentiPair Sequence to model the image sequence in a GIF. It contains over 40,000 GIFs which were manually labelled.

Since, we are dealing with strictly ‘single image per file’ data; we did not find this dataset suitable for our research.

CHAPTER 3
Research Methodology

This chapter first describes the types of attributes we have used for mining the visual data. In the next segment, the framework used for analysing sentiment in the images is described. In the last segment, some novel concepts and unfamiliar terms used in this research are described in detail.

3.1 DERIVING MID LEVEL ATTRIBUTES

Our main aim is to somehow reduce the semantic gap between the low level features of an image and the high level human sentiment features. To achieve that, we need to create some mid level features that act as a semantic bridge. We utilised two methods to create mid level features in this research: creating image content based features and extracting attached tags of the uploaded images.

Content based visual features take advantage of the actual visible attributes of an image like brightness, symmetry, color ratio, edges, curves, dimensionality etc. These features automatically provide a way to describe the images uniquely. Also, as they are inherently a part of the image itself, they provide us with a true and genuine property of the image through which it can be measured. There is no possibility of creating false attributes. These features are retrieved with the help of various image filters assimilated with Weka.

Another way to create mid level attributes is to look up for the annotations typed with the images. When we upload images on any social media, we are provided with the option to write tags, comments or annotations with the image. These annotations provide a useful insight into the meaning of the image. By retrieving such tags, we can reduce the semantic gap and create mid level features.

3.2 OVERVIEW OF THE FRAMEWORK

A summarization of the whole process is shown in Figure 3.1.

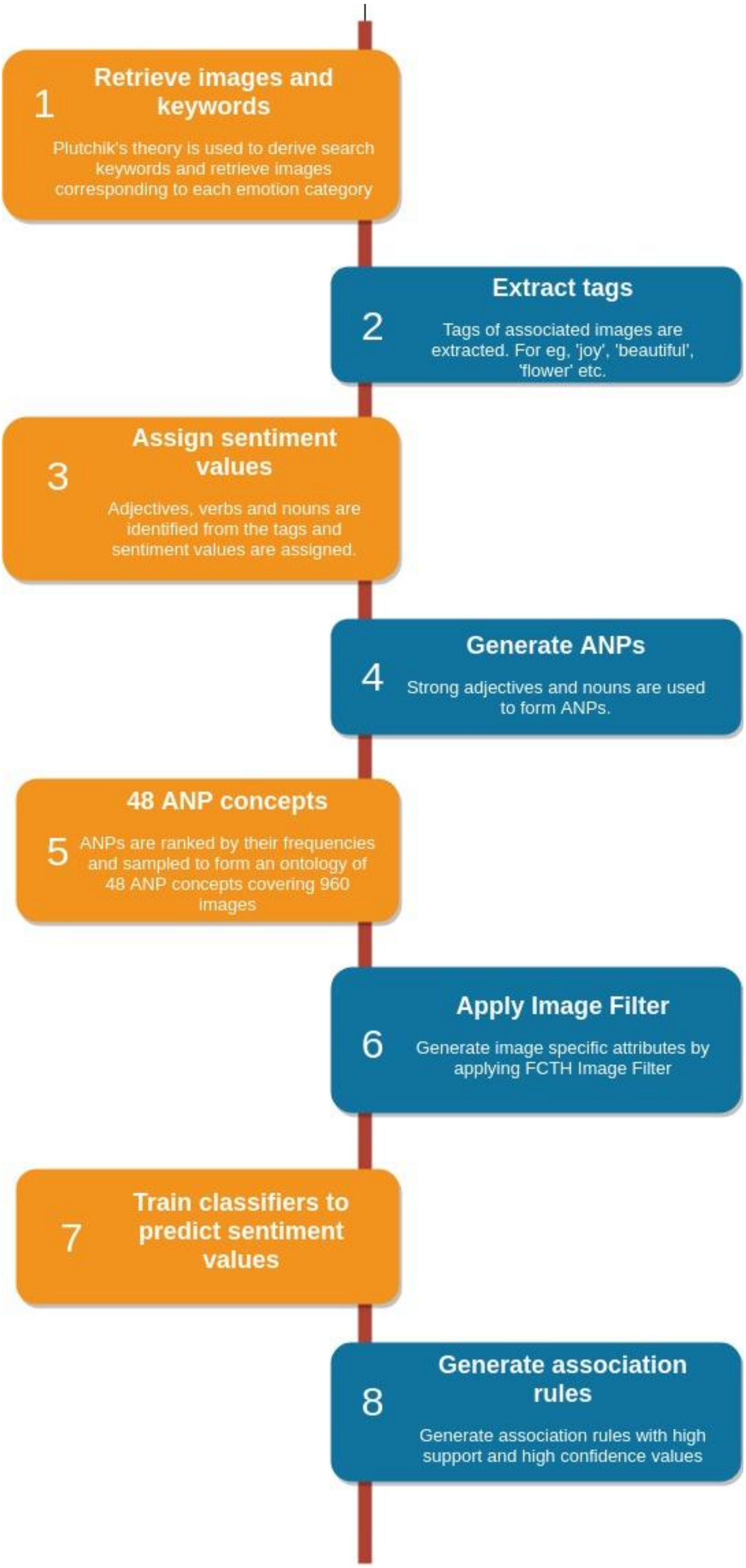


Figure 3.1: Approach overview

Step 1: Retrieve images and keywords

The foremost job is to create a dataset which means retrieving images from the Internet. In this research, we have manually selected 960 images from Flickr. The images selected corresponded to the eight emotions of the Plutchik's Wheel of Emotions. For each emotion category, 120 images are downloaded. For each image, related keywords are also retrieved.

Step 2: Extract tags

All the associated keywords, retrieved alongside the images, are analysed. Stopwords removal and stemming is performed. Then, frequency analysis is done to extract tags with significant frequency.

Step 3: Identify adjectives, verbs and nouns

From the top tags, adjectives, verbs and nouns are identified manually. And for each tag, a sentiment value is assigned.

Step 4: Generate ANPs

Strong adjectives and nouns pairs are formed to generate ANPs. Pairs of adjectives and nouns are formed because only adjectives or only nouns are unable to convey strong emotions on their own. Also, some adjectives and nouns totally change the emotion of the context when grouped together.

Step 5: 48 ANP concepts

All the ANPs formed are again ranked according to their frequencies and sampled to form ontology of 48 ANP concepts. There are six ANP concepts for each emotion.

Step 6: Apply Image Filter

Content based image features are retrieved using image filter. The filter applied is Weka FCTH Filter. It is a batch filter for extracting FCTH color features from images. FCTH stands for 'Fuzzy Color and Texture Histogram', and as the name suggests, these features encode both color and texture information in one histogram. One bonus of this feature is that it is very small, limited to 72 bytes per image, and therefore suitable for large image datasets [19].

Step 7: Train classifiers

Classifiers are trained to predict sentiment values. A variety of supervised classification techniques are used like SVM, logistic regression, Naive Bayes and K-nearest neighbour approach. A performance comparison analysis is done with the current research.

Step 8: Generate association rules

Apriori algorithm is used to generate association rules that predict sentiment category. Only those rules with high support and high confidence values are chosen.

3.3 GUIDE TO SOME NOVEL CONCEPTS USED

There are some terms and concepts used in this research that are not that common among general academia. A brief overview of these concepts is described below.

3.3.1 Plutchik's Wheel of Emotions

We have used Plutchik's theory as the psychological foundation of this research. His theory suggests that there are eight primary emotions of humans and each emotion can have three valences. This is shown in his wheel of emotions (Figure 3.2). Emotions that are contrast in nature are placed opposite to each other.

As shown in the figure, we have total eight emotions, each with three valences, described below:

1. ecstasy → joy → serenity
2. admiration → trust → acceptance
3. terror → fear → apprehension
4. amazement → surprise → distraction
5. grief → sadness → pensiveness
6. loathing → disgust → boredom
7. rage → anger → annoyance
8. vigilance → anticipation → interest [1]

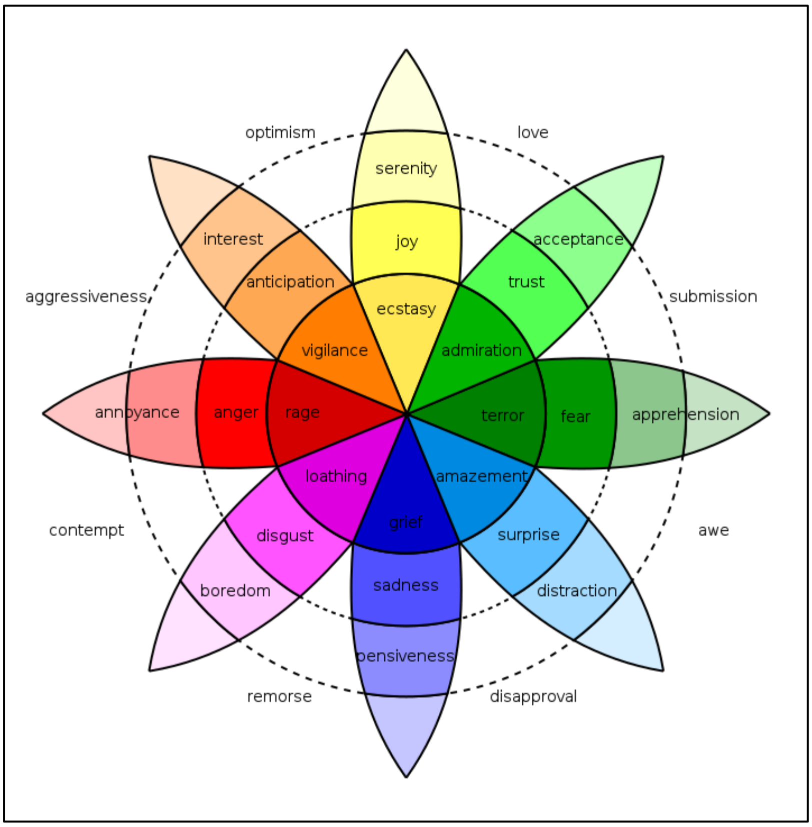


Figure 3.2: Plutchik's Wheel of Emotions [25]

This theory is used because it maps well to psychological theories. It depicts the emotions in a wheel like form and contrasting emotions are placed in opposite directions. It can also map to the old psychological theories like that of Ekman which had only six basic emotions, namely anger, disgust, fear, sadness, surprise, happiness. Ekman's 'happiness' corresponds to Plutchik's 'joy'. Plutchik also introduced two new emotions, admiration and interest. He organized each of the emotion in three valences which correspond to varying intensity of the same emotion. This theory suits well to our research as it is proved by the result accuracy.

3.3.2 Adjective Noun Pairs

A big hurdle in predicting sentiment of an image is to reduce the gap between low level and the high level features of an image. Adjective Noun Pairs, or simply ANPs, provide a mid level representation of the features.

With the image, several tags attached with it are also fetched from the Internet. After pre-processing these tags by performing stopwords removal and stemming, these tags are identified as adjectives, nouns, verbs, adverbs etc. Among these tags, it is seen that only one pair of adjective and noun is sufficient to demonstrate the content and underlying emotion of an image. One of the reasons for supporting the Adjective Noun Pair approaches is discussed here.

One might argue that only a single tag could also be used, whether adjective or noun, for the same purpose. But as evident from some examples, most of the adjectives and nouns when grouped together convey a totally different tone compared to when they are used individually. For example, ‘child’ is a positive noun, but when it is joined with a negative adjective like ‘abused’, the pair convey an intensely negative emotion.

Choosing the right ANPs is an important task in itself. Not all ANPs are popular on sites like Flickr. For example, ‘sweet flowers’ or ‘attractive cake’ does not sound satisfactory enough. ‘Sweet’ generally describes the taste of some food and ‘attractive’ is used in reference to beauty. Another problem arises when the ANPs are popular but the different adjectives are too similar with each other, like ‘cute kids’, ‘adorable kids’ or ‘lovely kids’. To remove ambiguity, only one of these ANPs should be considered and all the images under such tags should be placed in a single category.

3.3.3 FCTH Filter Features

To generate content based image features, which truly depict the genuine content specific attributes of an image, we need to apply some filter on the images that retrieve the physical properties of the images. This is achieved by applying the Fuzzy Color and Texture Histogram (FCTH) Filter provided under image filters of Weka.

The FCTH Filter extracts the color and texture information of an image which results from the application of three fuzzy systems and stores them in a single histogram that acts as a numeric feature of that image. This filter is highly suitable for large size datasets as it stores the features of a single image in a limited size of only 72 bytes. It can extract image features in case of distortions also and accurately fetch the data even when there is smoothing, noise or deformations.

CHAPTER 4
Proposed Solution

This chapter discusses the practical solution approach consideration of our thesis. It includes description of the manually constructed dataset, working of image filter, attributes ranking algorithm, classification techniques used and generated association rules.

4.1 DESCRIPTION OF THE DATASET

For this research, a large visual database consisting of images and tags taken from Flickr, is constructed manually. It consists of 960 images; each emotion has 120 images respectively.

The final top ranked ANPs selected for each emotion are as follows:

Joy: adorable cat, attractive flowers, bright rainbow, cute kids, happy smile, sweet cake

Sadness: bad accident, crying girls, lonely night, sad goodbye, sick dog, terrible tragedy

Disgust: dirty feet, dirty snow, fat pig, nasty bugs, rotten apple, sour milk

Fear: abandoned asylum, creepy doll, dark shadows, dead zombie, illegal war, scary spider

Interest: classical architecture, fancy hair, favourite book, holy places, rich history, talented student

Admiration: excellent museum, famous tower, fascinating places, gorgeous dress, magnificent church, smooth curves

Amazement: curious baby, excited face, incredible view, nice surprise, precious gift, wild party

Anger: angry bull, barking dog, mad cat, screaming face, violent crime, violent protest

For every given ANP, twenty images were sampled.

Apart from these, three additional tags were also considered for each image that effectively represents the image. These tags were chosen from the top ranked high frequency keywords. A list of the chosen tags is given in Figure 4.1.

```
sleepy,cute,pillow,kid,kitten,ears,paws,orange,tabby,bored,stunning,interest,beautiful,
eyes,black,rough,stray,baby,yellow,blue,breed,charming,fluffy,curious,white,whiskers,
pussy,yawn,green,plants,summer,pink,growth,buds,soft,dew,bloom,sun,rose,morning,pollen,
branch,red,trees,garden,leafy,pot,colorful,town,sky,rain,violent,scarf,wool,stripes,beads,
flowers,band,cake,tasty,circle,lights,festival,night,candy,sweet,castle,hill,city,cloudy,
paint,light,curves,shiny,smile,happy,child,snow,protest,naughty,silly,jungle,india,girls,
river,mother,nose,innocent,hair,drinks,party,fun,old,sea,click,lady,dogs,park,winter,laugh,
man,teeth,dress,car,puppy,country,nightmare,bullets,wedding,delicious,food,pastry,
decoration,recipe,chocolate,strawberry,slice,brownie,cupcake,spoon,window,smash,broken,
road,help,police,vehicle,smoke,injured,shadows,fire,firefighter,tyre,mud,stairs,ticket,
heartbreak,emotions,tears,camera,acting,news,family,gift,school,japan,friends,sisters,bond,
together,fight,street,walk,gloomy,dracula,person,umbrella,shoes,alone,bench,waiting,teddy,
bus,letter,grief,boat,evening,water,dark,airport,baggage,hug,love,despair,army,soldiers,
couple,grandma,cycle,handmade,balloon,wife,hope,birds,death,funeral,hospital,sad,ill,doctor,
surgery,rest,cold,oxygen,toy,better,spilled,crying,memory,parade,rich,murder,abandoned,
icecream,power,poison,laptop,faces,gold,fallen,shock,candles,priest,vietnam,gallery,history,
sculpture,stone,animals,knots,rope,goat,airplane,greek,elephant,painting,famous,cats,smelly,
messy,brain,ring,creepy,doll,bowl,lake,tall,bridge,big,leaning,pisa,eiffel,clock,skyscraper,
huge,dubai,shanghai,architecture,classical,elegant,church,building,london,modern,hands,empty,
museum,house,interior,ship,island,people,sunset,beach,vintage,cool,fashion,gown,tower,
religious,roof,good,carpet,turn,glasses,pumpkin,oil,blood,clown,halloween,zoo,tiger,grandpa,
favorite,book,hat,excited,door,surprise,dance,nature,temple,peace,cave,sand,birthday,ghost,
stars,robot,crazy,music,mad,hit,wild,run,outdoor,pet,fear,dirty,meow,angry,annoyed,pissed,
wet,pillar,king,scary,fat,gangster,gun,money,handcuffs,guilty,criminal,destroyed,trespass,
terrorist,guard,democracy,flags,rights,bomb,tank,dry,indoor,mix,pork,farm,snout,insect,gross,
centipede,cockroach,slug,slimy,weird,legs,giant,mosquito,spider,web,spoiled,fungus,fruit,bugs,
inedible,cheese,yuck,windmill,unique,ancient,style,places,wig,choice,letters,reading,library,
mystery,shelf,busy,collection,tradition,foreign,event,teacher,maths,focus,stage,master
```

Figure 4.1 Additional tags chosen

So, with each image, we have an adjective noun pair along with three additional tags attached with it.

4.1.1 Image Correction

The images downloaded were all of JPEG format with dimensions roughly around 640x640 pixels. Yet, all these images did not have similar color components or raster bands. To create a standard color format, we applied image correction by running them through a Java code that helps standardize the images to be accepted by further machine learning algorithms.

A snippet of the code used is shown in Figure 4.2.

```java
    static BufferedImage readImage(InputStream stream) throws IOException {
        Iterator<ImageReader> imageReaders = ImageIO
                .getImageReadersBySuffix("jpg");
        ImageReader imageReader = imageReaders.next();
        ImageInputStream iis = ImageIO.createImageInputStream(stream);
        imageReader.setInput(iis, true, true);
        Raster raster = imageReader.readRaster(0, null);
        int w = raster.getWidth();
        int h = raster.getHeight();

        BufferedImage result = new BufferedImage(w, h,
                BufferedImage.TYPE_INT_RGB);
        int rgb[] = new int[3];
        int pixel[] = new int[3];
        for (int x = 0; x < w; x++) {
            for (int y = 0; y < h; y++) {
                raster.getPixel(x, y, pixel);
                int Y = pixel[0];
                int CR = pixel[1];
                int CB = pixel[2];
                toRGB(Y, CB, CR, rgb);
                int r = rgb[0];
                int g = rgb[1];
                int b = rgb[2];
                int bgr = ((b & 0xFF) << 16) | ((g & 0xFF) << 8) | (r & 0xFF);
                result.setRGB(x, y, bgr);
            }
        }
        return result;
    }

    private static void toRGB(int y, int cb, int cr, int rgb[]) {
        float Y = y / 255.0f;
        float Cb = (cb - 128) / 255.0f;
        float Cr = (cr - 128) / 255.0f;

        float R = Y + 1.4f * Cr;
        float G = Y - 0.343f * Cb - 0.711f * Cr;
        float B = Y + 1.765f * Cb;

        R = Math.min(1.0f, Math.max(0.0f, R));
        G = Math.min(1.0f, Math.max(0.0f, G));
        B = Math.min(1.0f, Math.max(0.0f, B));

        int r = (int) (R * 255);
        int g = (int) (G * 255);
        int b = (int) (B * 255);

        rgb[0] = r;
        rgb[1] = g;
        rgb[2] = b;
    }
```

Figure 4.2 Image correction code

This code first reads a JPEG file and transforms it to a BufferedImage. It also checks to see if the input stream needs to be stripped down in bit depth. It tries to copy all the raster samples for the pixel at x, y of the input image into the RGB pixel array. In general, it converts the sample images from YCbCr to RGB Color Space.

4.1.2 ARFF File Format

To import the images and tags in the Weka tool, they are converted into ARFF file format. ARFF (Attribute Relation File Format) is used, instead of CSV because of several reasons:

It is less memory intensive than CSV. Also, it is faster and better for analysis because it includes metadata about column headers.

In ARFF file, the images, tags, adjective-noun pairs and class are in the data section (@data). It is represented as seven columns:

@attribute filename string

@attribute tagA

@attribute tagB

@attribute tagC

@attribute adjective

@attribute noun

@attribute class {joy,sadness,fear,disgust,interest,admiration,amazement,anger}

'Filename' attribute is a string attribute which corresponds to the name of the image. Attributes like 'tagA', 'tagB', 'tagC', 'adjective' and 'noun' must follow with the list of string values that they can acquire. As shown in the above example, the attribute 'class' can take on any one of the eight values that is enclosed within braces following it.

An example of the data section is as follows:

j1img01.jpg, sleepy, cute, pillow, adorable, cat, joy

4.1.3 Average Image Value

For better visualization, the average reduced image of the images of each adjective noun pair was calculated using MATLAB. The MATLAB code snippet as well as a few examples of the reduced images related to the category 'admiration' is shown in Figure 4.3 and Figure 4.4 respectively.

```
for k = 1 : numberOfImages
    fullFileName = fullfile(folder, imageFiles(k).name);
    fprintf('About to read %s\n', fullFileName);
    thisImage=imread(fullFileName);
    [thisRows, thisColumns, thisNumberOfColorChannels] = size(thisImage);
    if k == 1
        sumImage = double(thisImage);
        rows1 = thisRows;
        columns1 = thisColumns;
        numberOfColorChannels1 = thisNumberOfColorChannels;
        theyreColorImages = numberOfColorChannels1 >= 3;
    else

        if rows1 ~= thisRows || columns1 ~= thisColumns
            thisImage = imresize(thisImage, [rows1, columns1]);
        end
        if thisNumberOfColorChannels == 3 && numberOfColorChannels1 == 1
            thisImage = rgb2gray(thisImage);
            theyreColorImages = false;
        elseif thisNumberOfColorChannels == 1 && numberOfColorChannels1 == 3
            thisImage = cat(3, thisImage, thisImage, thisImage);
            theyreColorImages = true;
        end
        sumImage = sumImage + double(thisImage);
        if theyreColorImages
            displayedImage = uint8(sumImage / k);
        else
            displayedImage = sumImage;
        end
        imshow(displayedImage, []);
        drawnow;
    end
end

sumImage = uint8(sumImage / numberOfImages);
cla;
imshow(sumImage, []);
```

Figure 4.3 MATLAB code for calculating average image

Figure 4.3 shows a code that places RGB images in a folder and then averages them together to calculate a single mean image. If the input images are of different dimensions, it first resizes all the images to the size of the first input image. Also, if the images belong to different color bits, it converts them into the color profile of the first image. Finally, it computes and displays the final image.

Figure 4.4 is divided into several parts depicting the output mean image of different ANPs under the sentiment class ‘admiration’.

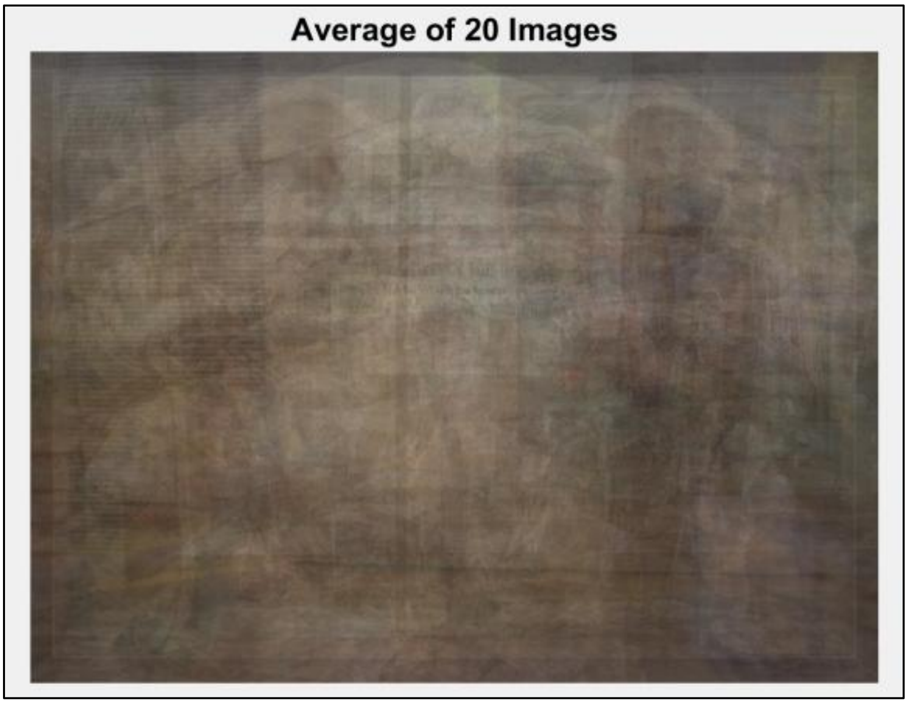


Figure 4.4a Excellent museum

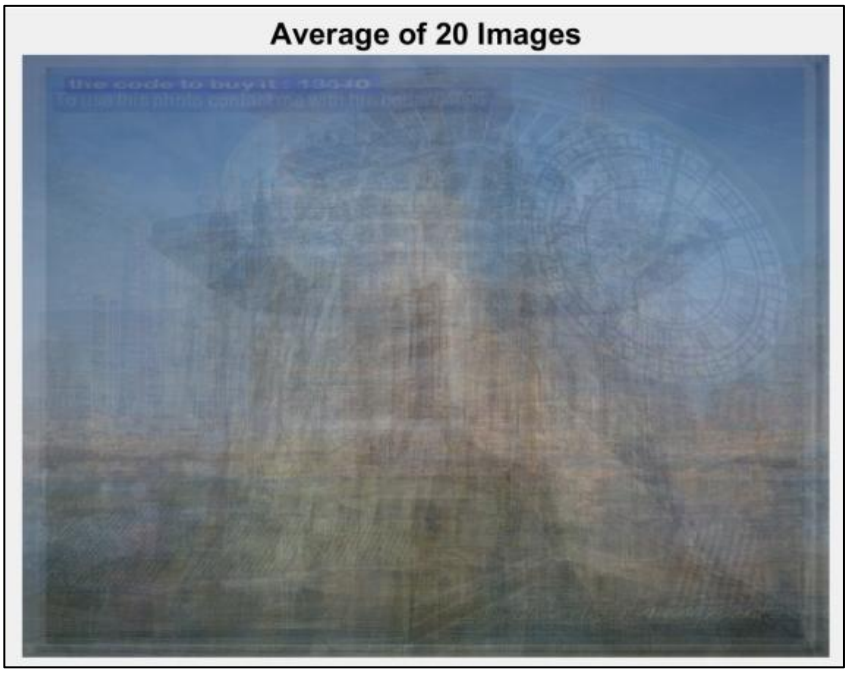


Figure 4.4b Famous tower

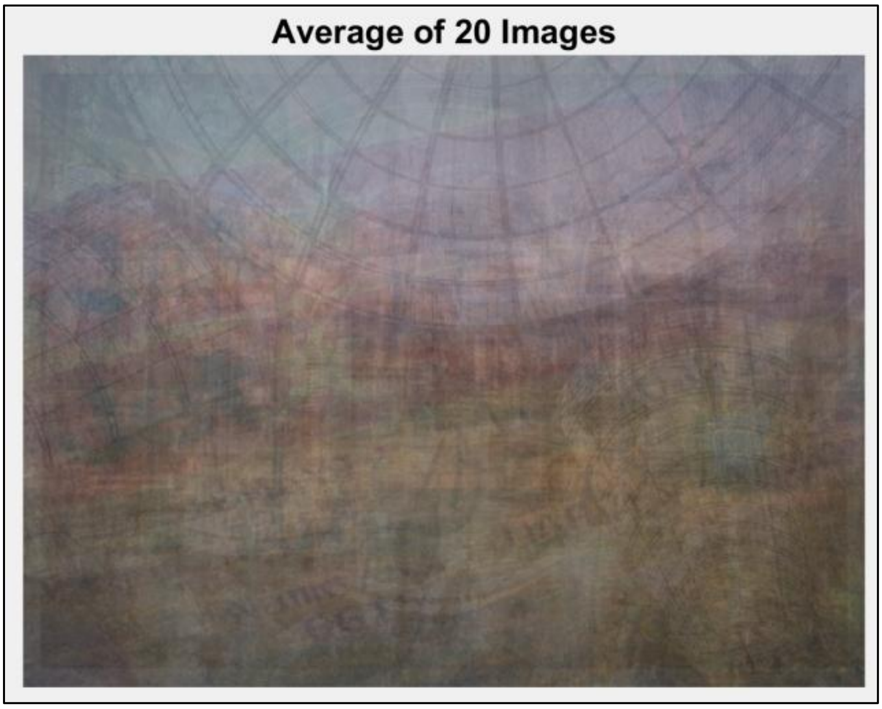


Figure 4.4c: Fascinating places

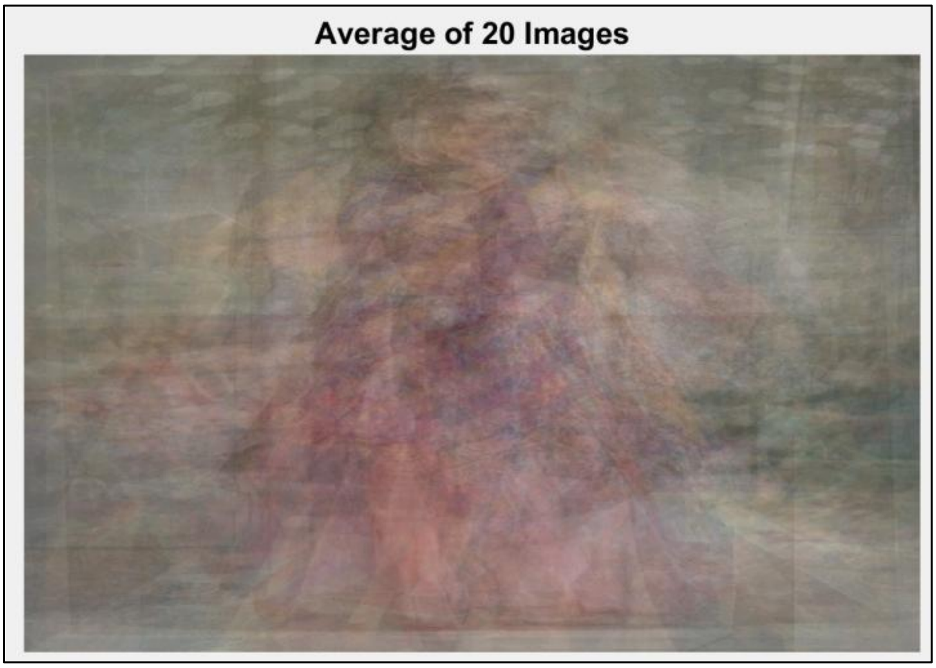


Figure 4.4d: Gorgeous Dress

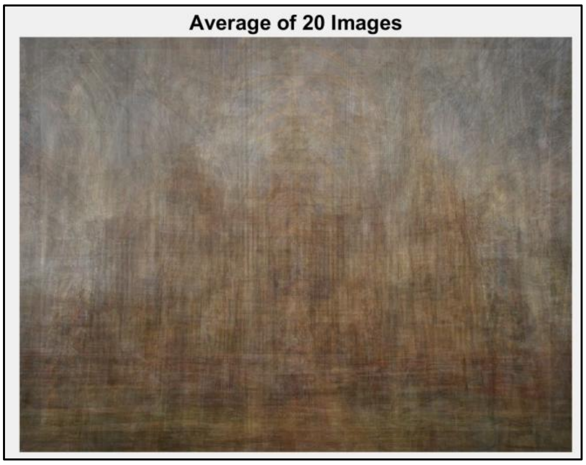


Figure 4.4e: Magnificent church

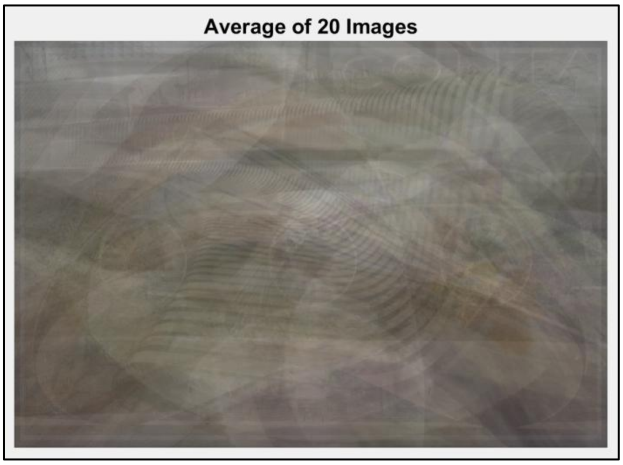


Figure 4.4f: Smooth curves

4.2 WEKA IMAGE FILTER

After importing the ARFF file to Weka, Weka allows us to visualize the class frequency distribution of the uploaded images. In Figure 4.5, we can see that there are 120 images (equal frequency) for each category of sentiment, depicted by different colors. It also specifies that there are total 960 instances and 7 attributes. One of them is the 'filename' attribute which has to be removed later as it is a string and that will cause problems for many different classifiers. Weka's 'Selected Attribute' window also shows that the class attribute has eight values and all of them are distinct. Weka provides us with the count and weight of each attribute. It should be noted that each class is given equal weightage because of equal frequency.

In Figure 4.6, the count frequency of 'tagA' is shown. 'Selected Attribute' window provides the details of this attribute. It shows 'tagA' has 291 distinct values and each value is automatically assigned a different weight corresponding to its frequency. A value that appears frequently will have more weight than a value that appears once or twice. And rightly so, as a value that appears frequently can more effectively help predicting a class than a less frequent value.

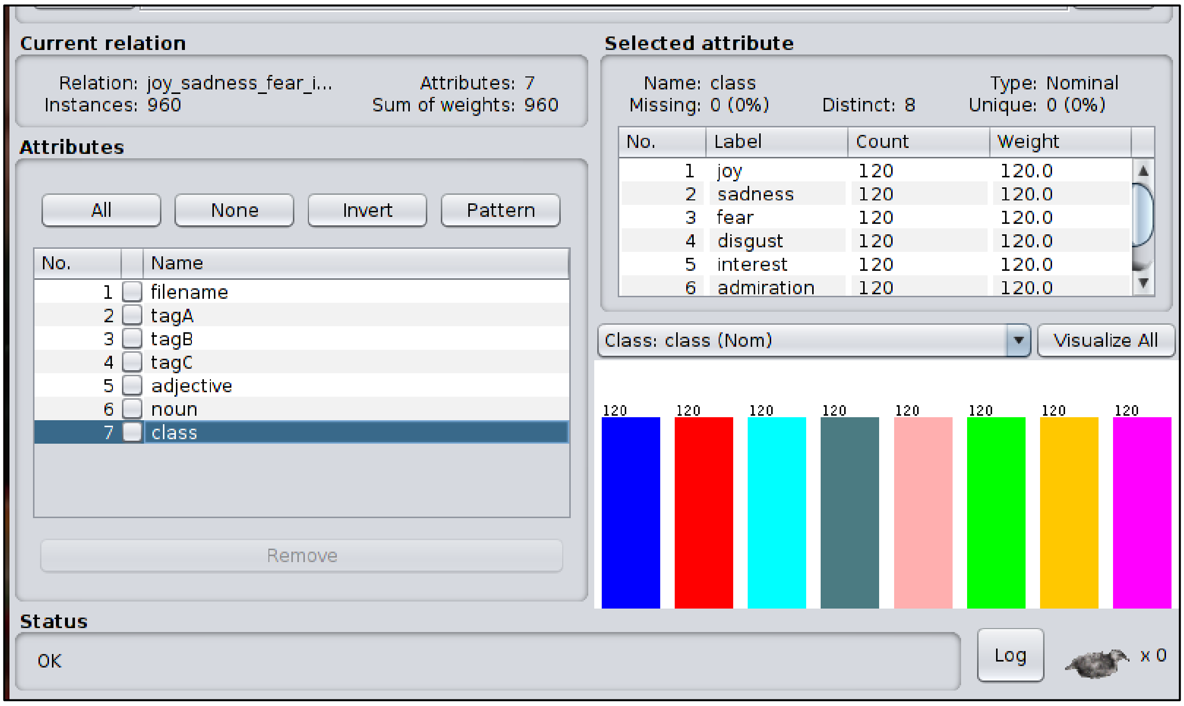


Figure 4.5 Class frequency for each sentiment

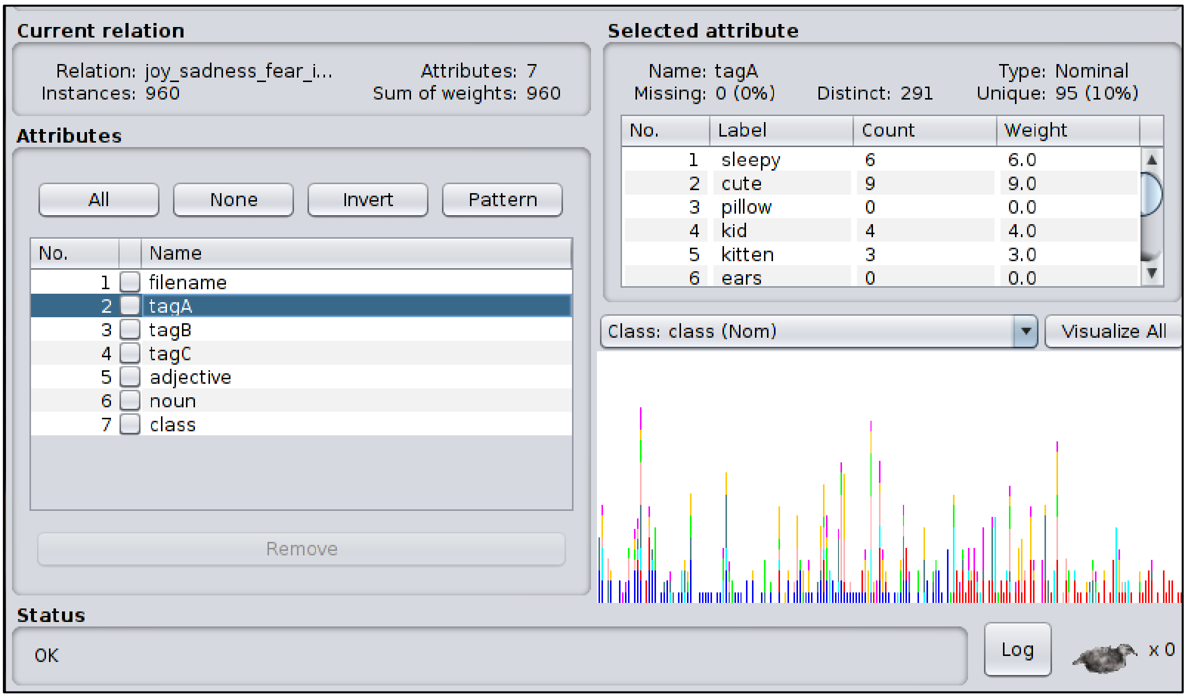


Figure 4.6 Count frequency for tagA

But these are only the features that are extracted along side images from Flickr. We need to generate content based image specific features too. For that, FCTH Filter provided by Weka for image classification is used. This filter comes under Image Filter Package for Weka and can be downloaded with Weka Package Manager.

FCTH Filter lets us convert images into features that can run image classificatioon experiments. Basically, image features are measurements concerning an image. It is a way of describing an image in terms of its color, texture and shape properties. Once, we calculate those measurements, we can put them together in a feature vector, and then we can use Weka's standard machine learning algorithms to do some image experiments and see if we can classify images with good accuracy.

When we apply the filter, it is going to add further attributes to the dataset. These are numeric attributes and a list of these attributes is shown in Figure 4.7.

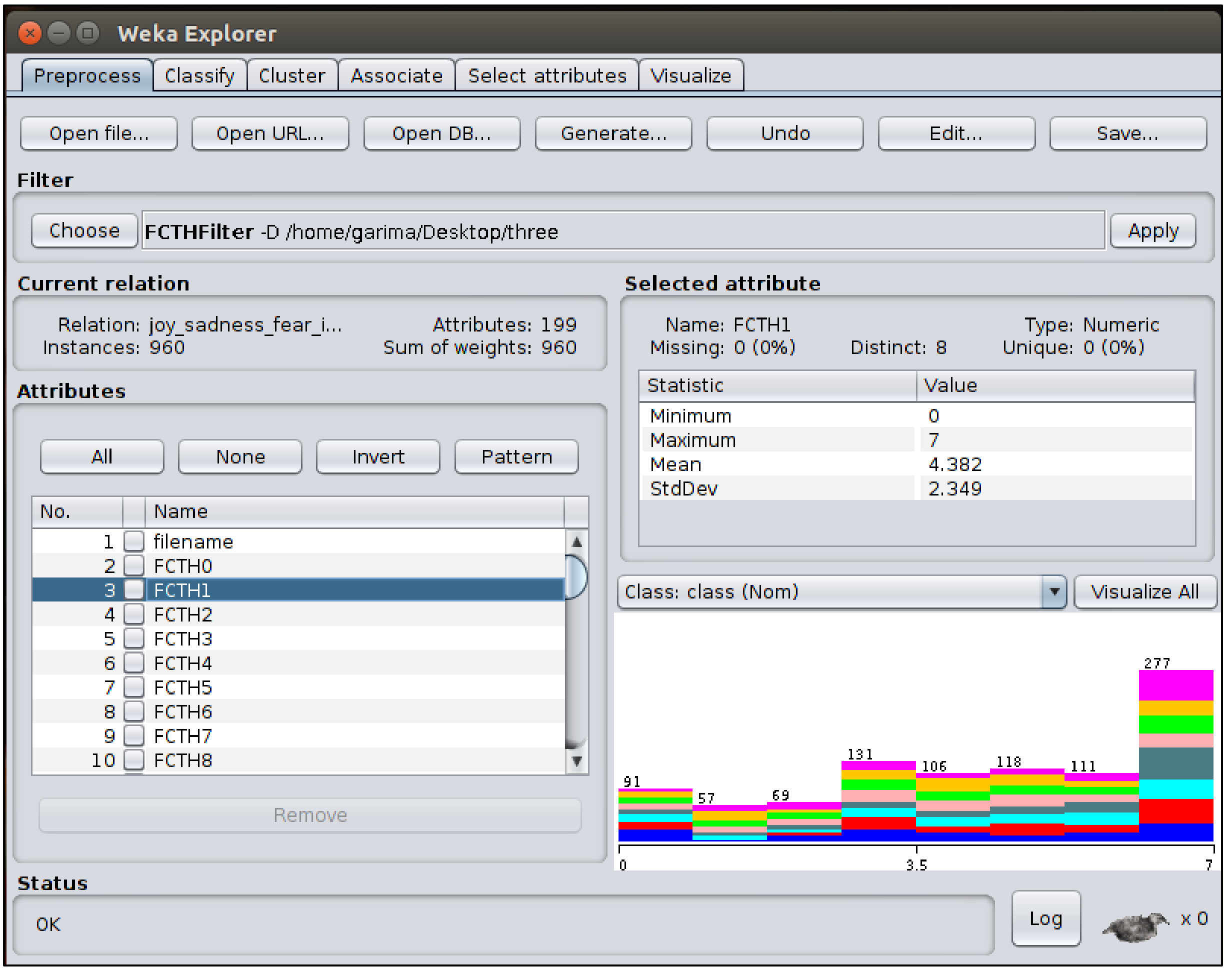
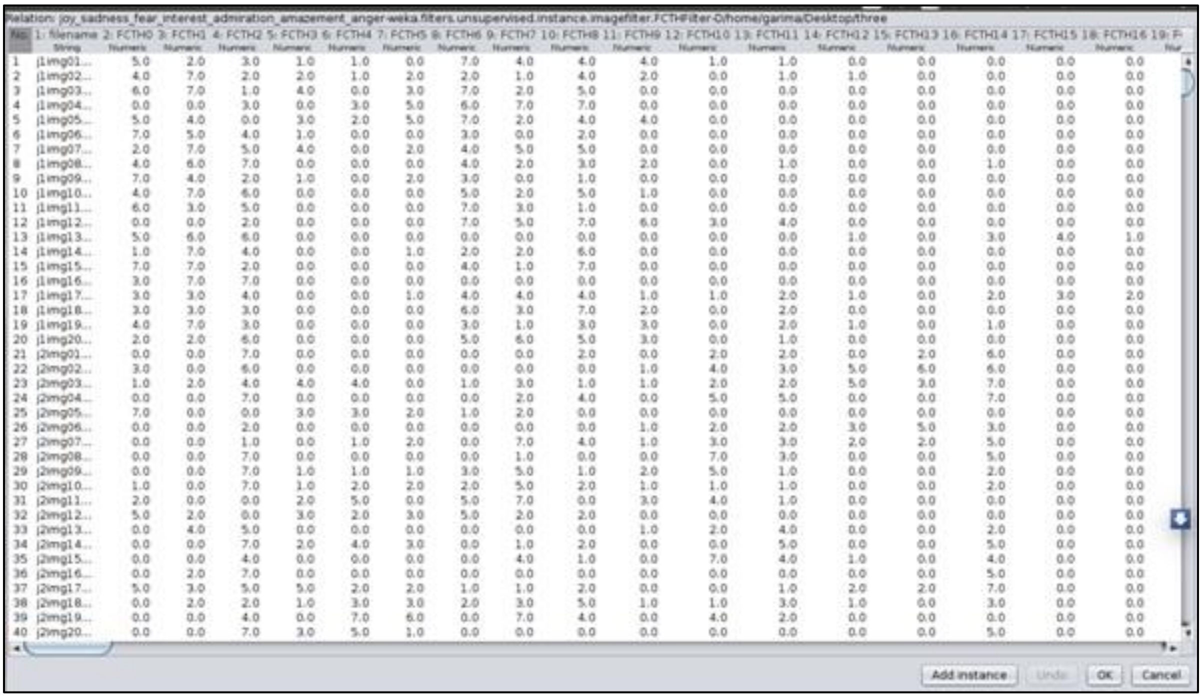


Figure 4.7 FCTH Filter attributes

In our case, FCTH Filter generated 192 additional features. ‘Select Attribute’ window give details about the minimum, maximum value of that feature along with mean and standard deviation. A glimpse of the numeric values of some features is given in Figure 4.8.


Relation: joy_sadness_fear_interest_admiration_amazement_anger-weka.filters.unsupervised.instance.imagefilter.FCTHFilter-D/home/garima/Desktop/three

No.	1: filename	2: FCTH0	3: FCTH1	4: FCTH2	5: FCTH3	6: FCTH4	7: FCTH5	8: FCTH6	9: FCTH7	10: FCTH8	11: FCTH9	12: FCTH10	13: FCTH11	14: FCTH12	15: FCTH13	16: FCTH14	17: FCTH15	18: FCTH16
	String	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric
1	j1img01...	5.0	2.0	3.0	1.0	1.0	0.0	7.0	4.0	4.0	4.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
2	j1img02...	4.0	7.0	2.0	2.0	1.0	2.0	2.0	1.0	4.0	2.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0
3	j1img03...	6.0	7.0	1.0	4.0	0.0	3.0	7.0	2.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	j1img04...	0.0	0.0	3.0	0.0	3.0	5.0	6.0	7.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	j1img05...	5.0	4.0	0.0	3.0	2.0	5.0	7.0	2.0	4.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	j1img06...	7.0	5.0	4.0	1.0	0.0	0.0	3.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	j1img07...	2.0	7.0	5.0	4.0	0.0	2.0	4.0	5.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	j1img08...	4.0	6.0	7.0	0.0	0.0	0.0	4.0	2.0	3.0	2.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0
9	j1img09...	7.0	4.0	2.0	1.0	0.0	2.0	3.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	j1img10...	4.0	7.0	6.0	0.0	0.0	0.0	5.0	2.0	5.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	j1img11...	6.0	3.0	5.0	0.0	0.0	0.0	7.0	3.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	j1img12...	0.0	0.0	2.0	0.0	0.0	0.0	7.0	5.0	7.0	6.0	3.0	4.0	0.0	0.0	0.0	0.0	0.0
13	j1img13...	5.0	6.0	6.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	3.0	4.0	1.0
14	j1img14...	1.0	7.0	4.0	0.0	0.0	1.0	2.0	2.0	6.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	j1img15...	7.0	7.0	2.0	0.0	0.0	0.0	4.0	1.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	j1img16...	3.0	7.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
17	j1img17...	3.0	3.0	4.0	0.0	0.0	1.0	4.0	4.0	4.0	1.0	1.0	2.0	1.0	0.0	2.0	3.0	2.0
18	j1img18...	3.0	3.0	3.0	0.0	0.0	0.0	6.0	3.0	7.0	2.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0
19	j1img19...	4.0	7.0	3.0	0.0	0.0	0.0	3.0	1.0	3.0	3.0	0.0	2.0	1.0	0.0	1.0	0.0	0.0
20	j1img20...	2.0	2.0	6.0	0.0	0.0	0.0	5.0	6.0	5.0	3.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
21	j2img01...	0.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	2.0	2.0	0.0	2.0	6.0	0.0	0.0
22	j2img02...	3.0	0.0	6.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	4.0	3.0	5.0	6.0	6.0	0.0	0.0
23	j2img03...	1.0	2.0	4.0	4.0	4.0	0.0	1.0	3.0	1.0	1.0	2.0	2.0	5.0	3.0	7.0	0.0	0.0
24	j2img04...	0.0	0.0	7.0	0.0	0.0	0.0	0.0	2.0	4.0	0.0	5.0	5.0	0.0	0.0	7.0	0.0	0.0
25	j2img05...	7.0	0.0	0.0	3.0	3.0	2.0	1.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
26	j2img06...	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	2.0	2.0	3.0	5.0	3.0	0.0	0.0
27	j2img07...	0.0	0.0	1.0	0.0	1.0	2.0	0.0	7.0	4.0	1.0	3.0	3.0	2.0	2.0	5.0	0.0	0.0
28	j2img08...	0.0	0.0	7.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	7.0	3.0	0.0	0.0	5.0	0.0	0.0
29	j2img09...	0.0	0.0	7.0	1.0	1.0	1.0	3.0	5.0	1.0	2.0	5.0	1.0	0.0	0.0	2.0	0.0	0.0
30	j2img10...	1.0	0.0	7.0	1.0	2.0	2.0	2.0	5.0	2.0	1.0	1.0	1.0	0.0	0.0	2.0	0.0	0.0
31	j2img11...	2.0	0.0	0.0	2.0	5.0	0.0	5.0	7.0	0.0	3.0	4.0	1.0	0.0	0.0	0.0	0.0	0.0
32	j2img12...	5.0	2.0	0.0	3.0	2.0	3.0	5.0	2.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
33	j2img13...	0.0	4.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	2.0	4.0	0.0	0.0	2.0	0.0	0.0
34	j2img14...	0.0	0.0	7.0	2.0	4.0	3.0	0.0	1.0	2.0	0.0	0.0	5.0	0.0	0.0	5.0	0.0	0.0
35	j2img15...	0.0	0.0	4.0	0.0	0.0	0.0	0.0	4.0	1.0	0.0	7.0	4.0	1.0	0.0	4.0	0.0	0.0
36	j2img16...	0.0	2.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0
37	j2img17...	5.0	3.0	5.0	5.0	2.0	2.0	1.0	1.0	2.0	0.0	0.0	1.0	2.0	2.0	7.0	0.0	0.0
38	j2img18...	0.0	2.0	2.0	1.0	3.0	3.0	2.0	3.0	5.0	1.0	1.0	3.0	1.0	0.0	3.0	0.0	0.0
39	j2img19...	0.0	0.0	4.0	0.0	7.0	6.0	0.0	7.0	4.0	0.0	4.0	2.0	0.0	0.0	0.0	0.0	0.0
40	j2img20...	0.0	0.0	7.0	3.0	5.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0

Add instance Undo OK Cancel

Figure 4.8 Numeric values of image features

4.3 RANKING ATTRIBUTES

The last pre-processing operation is the attribute selection and ranking. Eliminating the poorly characterizing attributes can be useful to get better classification accuracy. Also, the chosen attributes ranked in order of their classification efficiency helps us to judge the capacity of each attribute. For this task, Weka provides the ‘AttributeSelect’ filter from the weka.filters.supervised.attribute package. The filter allows choosing an attribute evaluation method and a search strategy.

Evaluator: These are the judges which judge the predictive quality of every attribute.

Search: These consult with the judge and make the final decision about which attributes are accepted and rejected.

The evaluator method and search strategy used in our experiment is discussed in detail in subsequent sections.

4.3.1 Chosen Evaluator: InfoGainAttributeEval

It evaluates the significance of an attribute by measuring the information gain with respect to the class. Information Gain (IG) is an entropy-based feature evaluation method, widely used in the field of machine learning. As Information Gain is used in feature selection, it is defined as the amount of information provided by the feature items for the text category. Information gain is calculated by how much of a term can be used for classification of information, in order to measure the importance of attributes for the classification. The formula of the information gain is shown in Equation 4.1.

$$InfoGain(Class,Attribute) = H(Class) - H(Class \mid Attribute) \qquad (4.1)$$

where H is entropy of the element.

4.3.2 Chosen Search: Ranker

It gives us a list of the attributes, ordered by their score according to the evaluator. It was selected with a zero threshold for information gain value. All the attributes which meet this threshold (i.e. attributes with positive information gain) are listed rank-wise.

Figure 4.9 shows the output of performing ‘AttributeSelect’ filter.

It used the full training set to rank the attributes. In the output, we can see that ‘adjective’ and ‘noun’ has the highest worth as attributes and they play a significant role in predicting the sentiment class of an image. After that, the three tags, ‘tagA’, ‘tagB’ and ‘tagC’ are valuable in deciding the underlying emotion of the image. In the end, the content specific FCTH filter features come into play in helping the decision process of assigning a sentiment class to the sampled image.

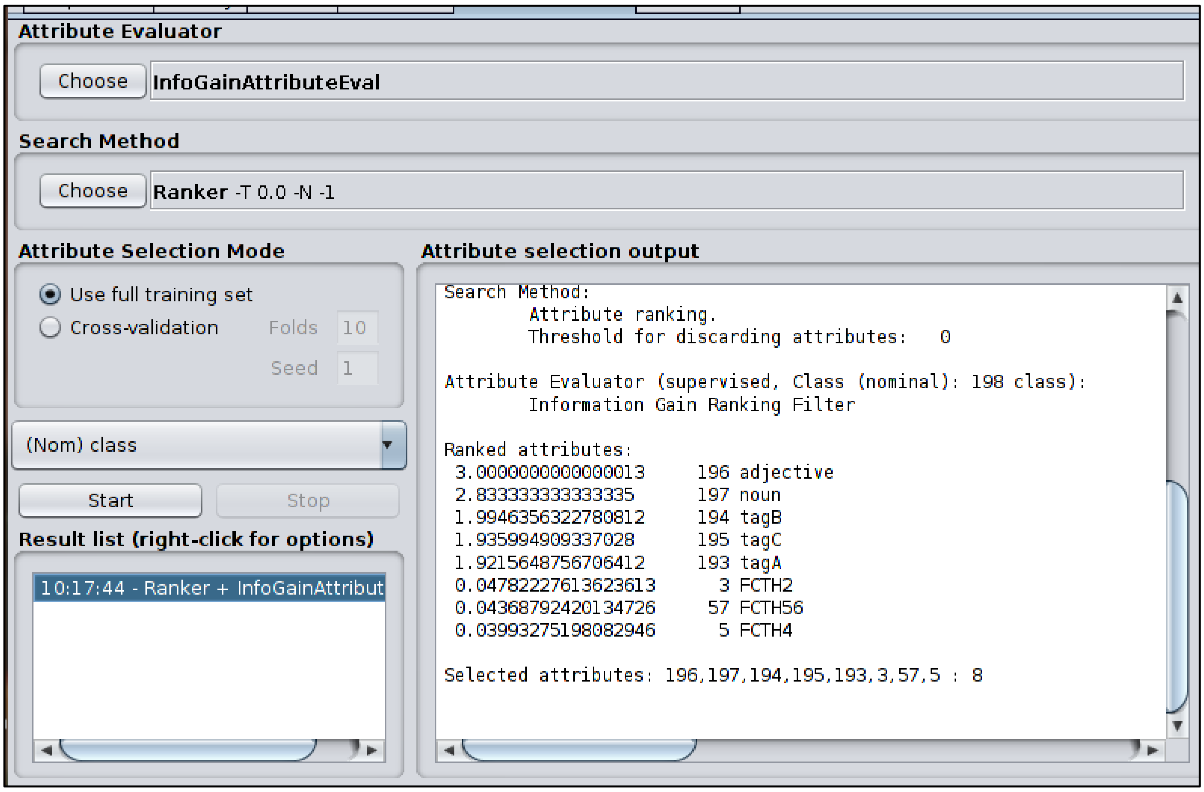


Figure 4.9 AttributeSelect Output

4.4 CLASSIFICATION

The classification task is a supervised learning method applied to a dataset. It incorporates designating a class label to an unclassified instance taking in consideration an already classified instance set, which works as a training set for the classification algorithm.

The classification algorithms that are executed in this project are Naive Bayes and K-nearest neighbors (k = 5). Some other algorithms like support vector machine and logistic regression are also executed to perform comparison analysis with the current research.

The quality measure that is considered is the percentage of correctly classified instances. For the validation (testing) phase, the 5-fold cross validation method is used.

The working procedure of the 5-fold cross validation method (for say 100 labelled data) is discussed below:

- Weka takes an input of 100 labelled data items.
- It generates 5 equal sized datasets. Each dataset is further split up into two groups: 90 labelled data items which are used for training purpose and 10 labelled data items which are used for testing purpose.

- It generates a classifier with the chosen algorithm from 90 labelled data items and applies that on the 10 testing data items for the first dataset.
- It performs the exact same thing for all the 5 datasets and generates 9 more classifiers.
- It calculates the average of the performance of the 5 classifiers generated from the 5 equal sized (90 training and 10 testing) datasets.

In the next subsections, the different classifiers used and their output is discussed in detail.

4.4.1 Naive Bayes

For each of the attribute, Naive Bayes assigns a probability score to all types of sentiments that contrasts how discriminative, or lack thereof, each attribute probably is. The Naive Bayes classifier designates a class to an unlabeled instance according to a maximum likelihood principle.

```
Classifier output

Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier

                           Class
Attribute                    joy   sadness      fear   disgust  interest admiration  amazement     anger
                          (0.13)    (0.13)    (0.13)    (0.13)    (0.13)    (0.13)    (0.13)    (0.13)
==========================================================================================================
FCTH0
  mean                     3.075      2.65     2.225       2.9       2.7    2.6667       2.6    2.7833
  std. dev.               2.3775    2.1666    2.4065    2.3502    2.2971    2.4198     2.181    2.2883
  weight sum                 120       120       120       120       120       120       120       120
  precision                    1         1         1         1         1         1         1         1

FCTH1
  mean                      3.95    4.6917    4.4333    5.0917     3.975    4.1583         4    4.7583
  std. dev.               2.4895    2.2833    2.3371    2.2472    2.2078    2.3274    2.2361    2.3629
  weight sum                 120       120       120       120       120       120       120       120
  precision                    1         1         1         1         1         1         1         1

FCTH2
  mean                     3.775     4.675     4.875    3.2583    4.2167      3.45      4.55       4.2
  std. dev.               2.2966    2.1415    2.3929    2.0636    2.0744    2.3233    2.2318     2.068
  weight sum                 120       120       120       120       120       120       120       120
  precision                    1         1         1         1         1         1         1         1

FCTH3
  mean                    0.9431    0.4861      0.35    0.4472    0.6028    0.4958    0.7194    0.4569
  std. dev.               1.3069    1.0169    0.7768     1.088    1.2419    1.0289    1.1087    0.9074
  weight sum                 120       120       120       120       120       120       120       120
  precision               1.1667    1.1667    1.1667    1.1667    1.1667    1.1667    1.1667    1.1667
```

Figure 4.10 Naive Bayes Output

Figure 4.10 shows how the Naive Bayes classifier has assigned a value to each attribute depicting how much they correspond to each emotion. The values assigned are numeric mathematic quantifiers like mean, standard deviation, weight sum and precision.

If the sentiment probabilities of an attribute are not far apart and it is used in the Naive Bayes algorithm's equation, that attribute will have less impact. Alternatively, if they are far apart, it will have more impact on the classification.

When the next image is fed to this classifier for classification, Naive Bayes will use the attributes in the image and compare the individual probabilities of all sides of the sentiment classes and then it will guess the most probable class.

It is called 'naive' as it learns about each attribute independently. Each attribute is scored in and of itself. It doesn't care how it is related to other attributes.

When used with our dataset, it worked with 69.3% efficiency as shown in Figure 4.11.

Classifier output

```
Correctly Classified Instances         665               69.2708 %
Incorrectly Classified Instances       295               30.7292 %
Kappa statistic                          0.6488
Mean absolute error                      0.0826
Root mean squared error                  0.2556
Relative absolute error                 37.7707 %
Root relative squared error             77.2795 %
Total Number of Instances              960

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.767    0.071    0.605      0.767    0.676      0.630    0.906     0.552     joy
                 0.750    0.056    0.657      0.750    0.700      0.656    0.861     0.673     sadness
                 0.708    0.031    0.766      0.708    0.736      0.701    0.870     0.733     fear
                 0.725    0.029    0.784      0.725    0.753      0.720    0.889     0.736     disgust
                 0.725    0.071    0.592      0.725    0.652      0.600    0.869     0.581     interest
                 0.500    0.033    0.682      0.500    0.577      0.535    0.842     0.574     admiration
                 0.700    0.036    0.737      0.700    0.718      0.679    0.877     0.735     amazement
                 0.667    0.024    0.800      0.667    0.727      0.696    0.855     0.743     anger
Weighted Avg.    0.693    0.044    0.703      0.693    0.692      0.652    0.871     0.666

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h   <-- classified as
 92  9  2  0  3  3  4  7 |  a = joy
  7 90  2  4  9  3  3  2 |  b = sadness
  7  6 85  6 10  3  3  0 |  c = fear
  8  4  3 87 10  4  4  0 |  d = disgust
  9  5  5  2 87  7  3  2 |  e = interest
 13  7  9  5 18 60  5  3 |  f = admiration
  6  9  2  4  4  5 84  6 |  g = amazement
 10  7  3  3  6  3  8 80 |  h = anger
```

Figure 4.11 Naïve Bayes Classification

Advantages of Naive Bayes algorithm:

- Using attributes as independent features makes it computationally efficient and relatively accurate.
- It works well on different types of data types.

- It provides a quick way to build models for high dimensional datasets which consists of many more columns than rows.

4.4.2 K-Nearest Neighbours

The k-nearest neighbours classifier assigns that class to an instance which is most frequent among the k nearest instances. For this to be attainable, a distance function must be defined between the instances. In this project, the Euclidean distance formula was used. The k parameter is chosen to be an odd number, so that a majority always exists.

In Weka, the k-NN classifier is implemented in the weka.classifiers.lazy.IBk component. The best result achieved with this kind of classifiers showed a correctness percentage of 88.9% (Figure 4.12), using the 5-nearest neighbours classifier i.e. k was assigned the value '5'. Search algorithm used iss LinearNNSearch.

Classifier output

```
Correctly Classified Instances         853               88.8542 %
Incorrectly Classified Instances       107               11.1458 %
Kappa statistic                          0.8726
Mean absolute error                      0.0296
Root mean squared error                  0.1662
Relative absolute error                 13.5387 %
Root relative squared error             50.2503 %
Total Number of Instances              960

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.942    0.018    0.883      0.942    0.911      0.899    0.962     0.839     joy
                 0.925    0.014    0.902      0.925    0.914      0.901    0.955     0.844     sadness
                 0.883    0.018    0.876      0.883    0.880      0.862    0.933     0.788     fear
                 0.933    0.020    0.868      0.933    0.900      0.885    0.957     0.819     disgust
                 0.842    0.014    0.894      0.842    0.867      0.849    0.914     0.772     interest
                 0.833    0.017    0.877      0.833    0.855      0.835    0.908     0.752     admiration
                 0.892    0.007    0.947      0.892    0.918      0.908    0.942     0.858     amazement
                 0.858    0.019    0.866      0.858    0.862      0.842    0.920     0.761     anger
Weighted Avg.    0.889    0.016    0.889      0.889    0.888      0.873    0.936     0.804

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   <-- classified as
 113   0   3   1   0   1   0   2 |   a = joy
   1 111   1   1   0   0   1   5 |   b = sadness
   3   1 106   4   1   3   0   2 |   c = fear
   0   2   1 112   2   1   2   0 |   d = disgust
   2   3   1   0 101   8   2   3 |   e = interest
   4   1   2   4   6 100   0   3 |   f = admiration
   2   1   4   3   1   1 107   1 |   g = amazement
   3   4   3   4   2   0   1 103 |   h = anger
```

Figure 4.12 K-Nearest Neighbours Classification

Weka also provides the output confusion matrix which demonstrates clearly how many instances of which category were classified correctly or incorrectly. Other than, accuracy percentage, it also provides us with quantifiable measurements like precision, recall and F-measure for each category of emotion.

4.4.3 Classification for comparison

Though our work was complete with Naive Bayes and k-Nearest Neighbours classifiers, but we need to execute some other classifiers also to compare our work with the existing research.

First of all, we have performed a text only classification which excludes ANPs and visual features generated by the image filter. It only considers the random tags that are fetched with the images from Flickr. The algorithm used is decision table. Its result is shown in Figure 4.13. As seen, it correctly classifies only 41% of the total instances.

Classifier output

```
Correctly Classified Instances         394               41.0417 %
Incorrectly Classified Instances       566               58.9583 %
Kappa statistic                          0.3262
Mean absolute error                      0.1964
Root mean squared error                  0.3052
Relative absolute error                 89.7905 %
Root relative squared error             92.2848 %
Total Number of Instances              960

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC     ROC Area  PRC Area  Class
                 0.633    0.317    0.222      0.633    0.329      0.219   0.757     0.385     joy
                 0.425    0.071    0.459      0.425    0.442      0.366   0.822     0.451     sadness
                 0.500    0.054    0.571      0.500    0.533      0.473   0.814     0.550     fear
                 0.575    0.044    0.651      0.575    0.611      0.560   0.837     0.609     disgust
                 0.267    0.063    0.376      0.267    0.312      0.237   0.724     0.293     interest
                 0.225    0.037    0.466      0.225    0.303      0.261   0.745     0.335     admiration
                 0.275    0.056    0.413      0.275    0.330      0.262   0.709     0.308     amazement
                 0.383    0.032    0.630      0.383    0.477      0.438   0.795     0.496     anger
Weighted Avg.    0.410    0.084    0.474      0.410    0.417      0.352   0.775     0.429

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h   <-- classified as
 76  6  7  7  6  5 11  2 |  a = joy
 49 51  4  4  3  1  4  4 |  b = sadness
 22 11 60  7  9  3  1  7 |  c = fear
 26  5  4 69  1  5  5  5 |  d = disgust
 41  4 12  3 32  9 14  5 |  e = interest
 44 10  4  7 17 27 10  1 |  f = admiration
 47 14  4  3 10  6 33  3 |  g = amazement
 37 10 10  6  7  2  2 46 |  h = anger
```

Figure 4.13 Decision table classification for text only attributes

This will be compared with the text only classification done by existing research in the next chapter. Some other classification variations that are also performed are discussed below.

In the recent research by Borth and Ji [1], they have compared their SentiBank approach with low-level representation by applying two algorithms: Linear SVM and Logistic Regression.

Therefore, we have also performed those two techniques to ease the process of comparison. Results are shown in Figure 4.14 and Figure 4.15.

Classifier output

```
Correctly Classified Instances         588               61.25   %
Incorrectly Classified Instances       372               38.75   %
Kappa statistic                          0.5571
Mean absolute error                      0.1958
Root mean squared error                  0.3045
Relative absolute error                 89.494  %
Root relative squared error             92.0806 %
Total Number of Instances              960

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.675    0.071    0.574      0.675    0.621      0.564    0.895     0.526     joy
                 0.725    0.060    0.635      0.725    0.677      0.629    0.883     0.535     sadness
                 0.617    0.054    0.622      0.617    0.619      0.565    0.884     0.524     fear
                 0.825    0.030    0.798      0.825    0.811      0.784    0.943     0.737     disgust
                 0.458    0.060    0.524      0.458    0.489      0.423    0.847     0.411     interest
                 0.450    0.075    0.462      0.450    0.456      0.379    0.787     0.334     admiration
                 0.525    0.056    0.573      0.525    0.548      0.487    0.829     0.414     amazement
                 0.625    0.038    0.701      0.625    0.661      0.617    0.896     0.567     anger
Weighted Avg.    0.613    0.055    0.611      0.613    0.610      0.556    0.871     0.506

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h   <-- classified as
 81  2  3  4  9  8 11  2 |  a = joy
  4 87  6  3  5  3  6  6 |  b = sadness
  2 10 74  5  7  9  1 12 |  c = fear
  6  1  3 99  0  3  2  6 |  d = disgust
  9  8 12  1 55 21 14  0 |  e = interest
 17  7  9  6 15 54  9  3 |  f = admiration
 19  9  1  4 12  9 63  3 |  g = amazement
  3 13 11  2  2 10  4 75 |  h = anger
```

Figure 4.14 SVM Classification

We can see that while Logistic Regression correctly classified 56.9% instances, SVM was able to classify 61.3% instances correctly.

SVM and Logistic regression both performs significantly less than Naïve Bayes and K-Nearest Neighbour classification but they hold a role in executing a performance analysis of all the methods.

We leave this section here at this point and will continue it in the next chapter where the worth of executing these additional classifiers will be visualized clearly in form of graphs.

```
Classifier output

Correctly Classified Instances         546               56.875  %
Incorrectly Classified Instances       414               43.125  %
Kappa statistic                          0.5071
Mean absolute error                      0.1315
Root mean squared error                  0.2738
Relative absolute error                 60.1021 %
Root relative squared error             82.8016 %
Total Number of Instances              960

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.583    0.075    0.526      0.583    0.553      0.487    0.877     0.546     joy
                 0.567    0.061    0.571      0.567    0.569      0.508    0.872     0.576     sadness
                 0.625    0.040    0.688      0.625    0.655      0.609    0.872     0.667     fear
                 0.750    0.037    0.744      0.750    0.747      0.711    0.945     0.823     disgust
                 0.525    0.077    0.492      0.525    0.508      0.435    0.828     0.496     interest
                 0.433    0.100    0.382      0.433    0.406      0.316    0.765     0.390     admiration
                 0.483    0.065    0.513      0.483    0.498      0.429    0.815     0.536     amazement
                 0.583    0.037    0.693      0.583    0.633      0.589    0.876     0.690     anger
Weighted Avg.    0.569    0.062    0.576      0.569    0.571      0.510    0.856     0.590

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h   <-- classified as
 70  5  1  1  5 15 20  3 |  a = joy
  7 68  8  4  9 10  7  7 |  b = sadness
  5  4 75  4 10 10  3  9 |  c = fear
  7  5  4 90  4  2  3  5 |  d = disgust
  9  4  8  5 63 22  9  0 |  e = interest
 13 11  5  9 20 52  7  3 |  f = admiration
 15 12  1  2 14 14 58  4 |  g = amazement
  7 10  7  6  3 11  6 70 |  h = anger
```

Figure 4.15 Logistic Regression Classification

4.5 ASSOCIATION RULES

Association rules are about finding associations between attributes. Rules can predict any attribute, or indeed, any number of attributes. For this, we need a different kind of algorithm. The one that we use in Weka, and the most popular association rule algorithm, is called Apriori. To discriminate between a set of rules that can be generated from a dataset, we use metrices like ‘support’ and ‘confidence’.

Support is the number of instances that satisfy a rule while confidence is the proportion of instances for which the conclusion holds. Typically, we set a minimum confidence value and look for rules with high support in that confidence range.

Apriori algorithm is used to determine data instances with high support and high confidence. It starts by determining the prevalent individual instances in the dataset and spans them to subsequently larger instances sets as long as those instances appear adequately often in the dataset. The most frequent instances identified by Apriori can be used to determine association rules which emphasise general trends in the dataset.

Figure 4.16 shows the output of Apriori algorithm, and some rules generated from that, along with the confidence and support values. We have considered only those rules that predict a class.

```
Best rules found:

 1. adjective=dirty 40 ==> class=disgust 40    conf:(1)
 2. adjective=violent 40 ==> class=anger 40    conf:(1)
 3. adjective=adorable 20 ==> class=joy 20    conf:(1)
 4. adjective=attractive 20 ==> class=joy 20    conf:(1)
 5. adjective=bright 20 ==> class=joy 20    conf:(1)
 6. adjective=cute 20 ==> class=joy 20    conf:(1)
 7. adjective=happy 20 ==> class=joy 20    conf:(1)
 8. adjective=sweet 20 ==> class=joy 20    conf:(1)
 9. adjective=bad 20 ==> class=sadness 20    conf:(1)
10. adjective=crying 20 ==> class=sadness 20    conf:(1)
```

Figure 4.16: Top 10 best rules

Some other rules with significant support and confidence are shown in Figure 4.17.

```
adjective=incredible noun=view 20 ==> class=amazement 20    conf:(1)
adjective=nice noun=surprise 20 ==> class=amazement 20    conf:(1)
adjective=precious noun=gift 20 ==> class=amazement 20    conf:(1)
adjective=wild noun=party 20 ==> class=amazement 20    conf:(1)
adjective=angry noun=bull 20 ==> class=anger 20    conf:(1)
adjective=barking noun=dog 20 ==> class=anger 20    conf:(1)
adjective=mad noun=cat 20 ==> class=anger 20    conf:(1)
adjective=screaming noun=face 20 ==> class=anger 20    conf:(1)
adjective=violent noun=crime 20 ==> class=anger 20    conf:(1)
adjective=violent noun=protest 20 ==> class=anger 20    conf:(1)
tagB=mad 15 ==> class=anger 15    conf:(1)
tagB=mud adjective=dirty 14 ==> class=disgust 14    conf:(1)
tagC=gross 11 ==> class=disgust 11    conf:(1)
tagB=spoiled 10 ==> class=disgust 10    conf:(1)
tagB=happy adjective=excited 10 ==> class=amazement 10    conf:(1)
tagB=happy noun=face 10 ==> class=amazement 10    conf:(1)
tagB=mad adjective=angry 10 ==> class=anger 10    conf:(1)
tagB=mad noun=bull 10 ==> class=anger 10    conf:(1)
tagB=happy adjective=excited noun=face 10 ==> class=amazement 10    conf:(1)
tagB=mad adjective=angry noun=bull 10 ==> class=anger 10    conf:(1)
tagB=tasty adjective=sweet 9 ==> class=joy 9    conf:(1)
```

Figure 4.17: Other significant rules

From Figure 4.16, we can note that the top rules with highest support and highest confidence are generated with the ‘adjective’ and ‘noun’ attributes only. The adjectives like ‘adorable’, ‘attractive’, ‘bright’, ‘cute’, ‘happy’ and ‘sweet’ are all likely to predict the ‘joy’ class i.e. the happiness emotion.

It should also be noted that ‘dirty’ and ‘violent’ adjectives correspond to ‘disgust’ and ‘anger’ class respectively with the highest support value as 40 and confidence value as 1, which means that there are 40 instances which satisfy this particular rule with 100% confidence.

When we stretch these rules to lowest threshold so that some tag values can also be generalised into rules, we come up with Figure 4.17. We can see that when the ‘tagA’ is ‘mad’, then the class predicted is ‘anger’ with a good support of 15 value, which also resonates well with the reasoning that ‘angry’ is very often used interchangeably as ‘mad’.

CHAPTER 5

Performance Comparison and Result

In this chapter, we finally executes an analysis comparing the latest existing research work with our own research. First, we perform a comparison on the basis of only two specific classifiers. Then, we compare on basis of only text attributes that is excluding tags and content specific image features. In the last, an analysis which demonstrates the actual comparison with all the features including tags, ANPs and image features is done. All these comparisons are visualized in forms of bar charts taking data from Chapter 2 which provides latest research information and Chapter 4 which provides our research information.

5.1 LINEAR SVM AND LOGISTIC REGRESSION COMPARISON

Borth and Ji compared their SentiBank approach with classification using low level features only. So we compared our work with both the low level feature classification and their classification using SentiBank approach in Figure 5.1.

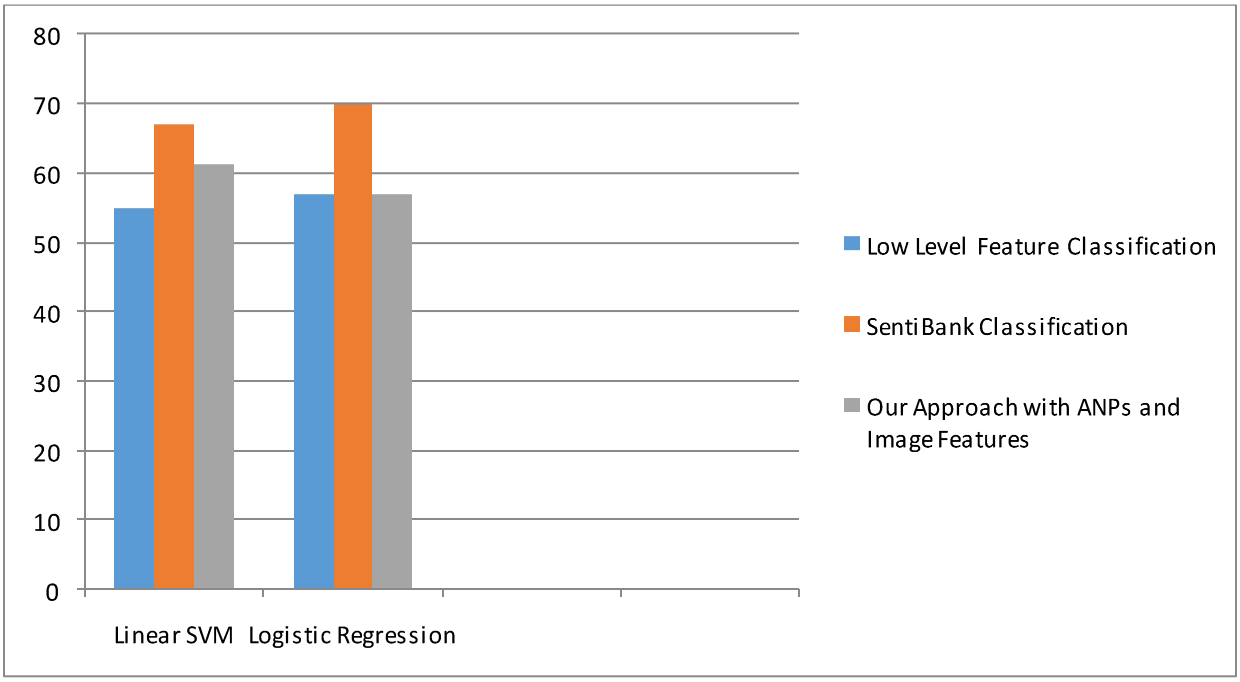


Figure 5.1 Comparison on basis of Linear SVM and Logistic Regression

Here, we can see that our approach is better in comparison with the classification based on low level features only. It lags behind in performance when compared to SentiBank approach. But this particular comparison was performed with only two supervised learning techniques, namely linear SVM and logistic regression. It was done only to demonstrate that our approach is better than the low level classification approach.

5.2 TEXT ONLY CLASSIFICATION COMPARISON

In this comparison, only text attributes were considered, which means ANPs and FCTH filter attributes were ignored. The graph is shown in Figure 5.2.

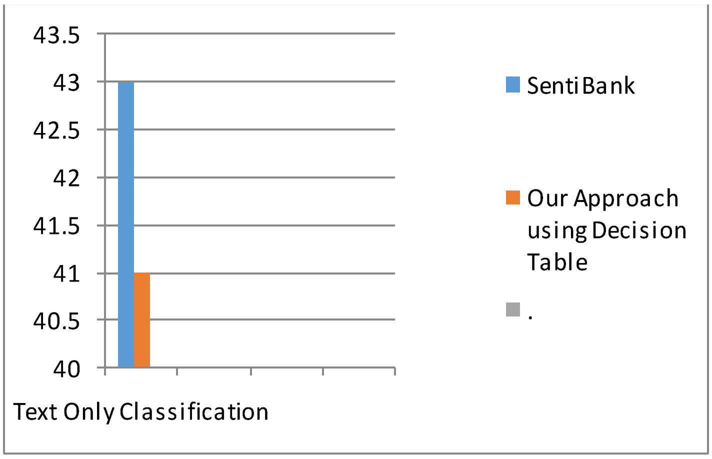


Figure 5.2 Text Only Classification

Since, it does not incorporate the ANPs and FCTH filter attributes, it does not let our approach to fully function in a suitable environment which leads to a much poor performance. What we need is to wholly incorporate our approach with mixed attributes to get its essence.

5.3 TEXT AND VISUAL ATTRIBUTES CLASSIFICATION

Our main research focuses on Naive Bayes and K-Nearest Neighbours models. So, a better comparison graph that rightly judges our research with the work done till now is shown in Figure 5.3. It also includes ANPs, FCTH filter attributes as well as all the tags.

This lets our method to function effectively and produces a very high performance index than the previous methods executed in recent research.

Figure 5.3: Text + Visual Attributes Classification

In Figure 5.3, the comparison has been performed on evaluation measures discussed above. It is a clear observation that our approach with K-NN (with ANPs, tags as well as image filter features) has the highest value among all other classification techniques where else SentiBank (with Linear SVM) has the least value.

This makes the K-nearest neighbour with ANPs as well as FCTH Filter features more applicable for accurate visual data sentiment prediction.

CHAPTER 6

Conclusion and Future Work

6.1 CONCLUSION

This project introduced and surveyed the field of sentiment analysis with respect to visual data. Due to many stimulating research problems and a wide variety of real life applications, it has been a bustling research area in the last few years. In fact, it has stretched out from computer science to management science.

The large visual dataset was manually constructed and labelled, comprising of 960 images from Flickr, 48 Adjective Noun Pairs and 291 tags.

Weka is a powerful tool in data mining. Techniques of Weka such as classification that is used to test and train different learning schemes on the pre-processed data file, image filters for retrieving image features and association rules algorithms were used to classify the data file into eight categories of sentiments and generate some rules with high support and high confidence value.

In this work, quality features were extracted that have a strong impact on determining the sentiment of the images. These features included tags extracted from the images uploaded on Flickr as well as content specific image features related to color and texture of the images. We applied various Weka filters for the pre-processing of the data. Weka also provided tools for modelling and visualization of the data. Then the feature impact analysis was performed by computing information gain for each feature in the feature set and used it to derive a reduced feature set. Contrasting the previously worked upon classification techniques like SVM and Logistic Regression used with SentiBank approach vs Naive Bayes and K-Nearest Neighbour used with our approach of mixed features, we found that the highest accuracy was given by K-Nearest Neighbour with mixed features with an accuracy of 88.9%.

We also generated association rules that give an insight into the sentiment class prediction of visual data. Apriori algorithm was used to generate association rules with high support and high confidence for this particular dataset.

6.2 FUTURE WORK

As a part of the future work, the algorithms can further be implemented in wide variety of datasets and then the result can be evaluated and compared. It will provide a comprehensive evaluation of our manually constructed dataset too. Also, it can be further improved to derive features from GIFs or even video data. This area is new and every effort put into it can produce significant results. Apart from that, face recognition projects can be combined with visual data sentiment prediction to further improve accuracy and make it effective in various real life applications.

References

[1] Damian Borth and Rongrong Ji, “Large-scale Visual Sentiment Ontology and Detectors Using Adjective Noun Pairs”, in MM’13, October 21–25, 2013, Barcelona, Spain. Copyright 2013 ACM 978-1-4503-2404-5/13/10.

[2] Rongrong JI and, Donglin CAO, “Survey of visual sentiment prediction for social media analysis”, in Higher Education Press and Springer-Verlag Berlin Heidelberg 2016, DOI 10.1007/s11704-016-5453-2.

[3] Brendan Jou and Tao Chen, “Visual Affect Around the World: A Large-scale Multilingual Visual Sentiment Ontology”, MM’15, October 26–30, 2015, Brisbane, Australia. 2015 ACM. ISBN 978-1-4503-3459-4/15/10. DOI: http://dx.doi.org/10.1145/2733373.2806246.

[4] Vani Kapoor Nijhawan, Mamta Madan , Meenu Dave, “T he Analytical Comparison of ID3 and C4.5 using WEKA”, International Journal of Computer Applications (0975 8887) Volume 167 No.11, June 2017.

[5] Tao Chen, Damian Borth, Trevor Darrell and Shih-Fu Chang, “DeepSentiBank: Visual Sentiment Concept Classification with Deep Convolutional Neural Networks”, arXiv:1410.8586v1 [cs.CV] 30 Oct, 2014.

[6] Soujanya Poria, Erik Cambria, Newton Howard, Guang-Bin Huang, Amir Hussain, “Fusing Audio, Visual and Textual Clues for Sentiment Analysis from Multimodal Content”, Neurocomputing, http://dx.doi.org/10.1016/j.neucom.2015.01.095

[7] Zheng Cai, Donglin Cao, Dazhen Lin, Rongrong Ji, “A Spatial-Temporal Visual Mid-Level Ontology for GIF Sentiment Analysis”, 978-1-5090-0623-6/16/$31.00 2016 IEEE.

[8] D. Borth, A. Ulges, and T.M. Breuel, “Lookapp – Interactive Construction of web-based Concept Detectors”, ICMR, 2011.

[9] R. Datta, D. Joshi, J. Li, and J. Wang., “Studying Aesthetics in Photographic Images using a Computational Approach”, ECCV, 2006.

[10] M. Everingham, et al. “The Pascal Visual Object Classes (VOC) Challenge. Int. J. of Computer Vision”, 88(2):303-338, 2010.

[11] A. Hanjalic, C. Kofler, and M. Larson. “Intent and its Discontents: the User at the Wheel of the Online Video Search Engine” ACM MM, 2012.

[12] Jianbo Yuan, Quanzeng You, Sean McDonough, and Jiebo Luo, "Sentribute: Image Sentiment Analysis from a Mid-level Perspective," ACM SIGKDD, Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM), Pages: 10:1--10:8, 2013

[13] Asur, S., and Huberman, B. A. 2010. "Predicting the future with social media", in WI-IAT, volume 1, 492–499. IEEE.

[14] Yilin Wang, Suhang Wang, Jiliang Tang, Huan Liu, and Baoxin Li, "Unsupervised Sentiment Analysis for Social Media Images", Arizona State University, Tempe, Arizona.

[15] Takuya Narihira et al, "Mapping Images to Sentiment Adjective Noun Pairs with Factorized Neural Nets", arXiv:1511.06838v1 [cs.CV] 21Nov2015.

[16] Monica, Jeremy, Jared, 2014. Handbook of Community Sentiment.

[17] www.ee.columbia.edu/ln/dvmm/vso/download/sentibank.html#download

[18] WekaMOOC Tutorial: More Data Mining with Weka" Published on Apr 10, 2014: https://youtu.be/iqQn6YfyGs0

[19] WekaMOOC Tutorial: Advanced Data Mining with Weka" Published on Apr 06, 2016: https://youtu.be/Lhw_XcGCTFg

[20] https://en.wikipedia.org/wiki/Weka_(machine_learning)

[21] https://developers.google.com/earth-engine/reducers_image_collection

[22] https://www.mathworks.com/matlabcentral/answers/23643-find-the-average-image-of-a-set-of-images

[23] https://expandedramblings.com/index.php/flickr-stats/

[24] https://gist.github.com/jpt1122/48903a7244b66ddecaf4

[25] https://en.wikipedia.org/wiki/Robert_Plutchik#/media/File:Plutchik-wheel.svg

[26] https://www.draw.io/