IDENTIFYING IMAGE RELATED SENTENCES IN NEWS ARTICLES

by

Melike Esma İlter

B.S., Computer Engineering, Yıldız Technical University, 2009

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2019

ACKNOWLEDGEMENTS

Firstly, with my deepest gratitude, I would like to thank my thesis supervisor Prof. Lale Akarun and my thesis co-supervisor Assoc. Prof. Arzucan Özgür for their expertise, support, guidance and patience during the research of this thesis. I would also like to extend my special thanks to my advisor Prof. Lale Akarun for her support throughout my master education program.

I am grateful and would like to thank Assoc. Prof. M. Elif Karslıgil and Prof. Tunga Güngör for participating in my thesis jury and their valuable comments on my thesis.

Finally, I would like to thank my family for their valuable and endless support, belief, love, encouragement and motivating me throughout my life while working hard for days and on sleepless nights. I could not make it without them. I am also grateful for my friends for their support, friendship and belief in me.

ABSTRACT

IDENTIFYING IMAGE RELATED SENTENCES IN NEWS ARTICLES

With the increasing availability of images on the web, identifying image related sentences has become an important problem. This research area is also important for the news publishing community for automatic captioning of news images and summarization. Although a large body of research has been devoted to image captioning, it is still a challenging problem. Previous works on image captioning mostly focus on generating new captions for the images. The problem of identifying image related sentences in news articles is discussed in this thesis for the first time and our approach is novel because we do not try to generate a caption from scratch, but we try to select the most appropriate set of sentences for the image from the news text itself. This technique helps not to lose the relationship between the news article and the image caption. We have used the CNN news dataset which only contains the text parts of news as basis and we have augmented the dataset by collecting the images of the news articles. We generated a two class ground truth for the image and sentences of news article by using Tf-Idf and Word2Vec vectors; and cosine and SEMILAR sentence-to-sentence similarity methods. We utilized HOG and BOVW image descriptors and Word2Vec text feature extraction methods. We implemented Naive Bayes, k-NN and Random Forest classification methods to measure the performance of our proposed system. We have also applied PCA dimensionality reduction method for image features to evaluate the equal weights of image and text features. We have also conducted experiments to solve the unbalanced class distribution of the two classes. The experiment results show that Naive Bayes classifier with HOG features gives better results.

ÖZET

HABER MAKALELERİNDE GÖRÜNTÜ İLE İLGİLİ CÜMLELERİN BELİRLENMESİ

İnternet ortamında görüntülerin ulaşılabilirliğinin artması ile görüntü ile ilgili cümlelerin belirlenmesi önemli bir problem haline gelmiştir. Bu araştırma alanı, haber görüntü altyazılarının otomatik oluşturulması ve özetleme konusunda haber yayıncıları için de önemlidir. Hakkında birçok çalışma yapılmış olsa da görüntü altyazılama hala zor bir problemdir. Görüntü altyazılama üzerine yapılmış önceki çalışmalar, genellikle görüntüler için yeni altyazılar üretmek üzerine odaklanmıştır. Haber metninden görüntü için en uygun cümleleri seçme problemi ilk defa bu çalışmada ele alınmıştır ve sıfırdan bir altyazı oluşturmaya çalışmak yerine görüntü ile ilgili cümleleri bulmaya çalıştığımız için yenidir. Bu teknik, haber ile resim yazısı arasındaki ilişkiyi kaybetmemeye yardımcı olur. Haberlerin sadece metin kısımlarını içeren CNN haber veri setini baz olarak kullandık ve bu veri setini haberlerin görüntülerini toplayarak genişlettik. Tf-Idf ve Word2Vec vektörleri kosinüs ve SEMILAR cümleden cümleye benzerlik yöntemlerini kullanarak haberin görüntüsü ve cümleleri için iki sınıflı referans altyapısı oluşturduk. HOG ve BOVW görüntü tanımlayıcıları ve Word2Vec metin özellik çıkarma yöntemlerini kullandık. Önerdiğimiz sistemin performansını ölçmek için Naive Bayes, k-En Yakın Komşu ve Rassal Karar Ormanı yöntemlerini uyguladık. Ayrıca, görüntü ve metin özelliklerini eşit ağırlıkla değerlendirmek için görüntü özellikleri için PCA boyut azaltma yöntemini uyguladık. Aynı zamanda, ikili sınıfların dengesiz dağılımını çözmek için de deneyler yaptık. Deney sonuçları, HOG özellik seçimi ile Naive Bayes sınıflandırıcısının daha iyi sonuçlar verdiğini göstermektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

C	Covariance matrix
H	Hessian matrix
$\widehat{X}$	A zero mean centered feature vector matrix
v	Eigenvector
W	Projection matrix
μ	PCA average vector
λ	Eigenvalue
Θ	Angle

LIST OF ACRONYMS/ABBREVIATIONS

BOVW	Bag of Visual Words
BOW	Bag of Words
CBOW	Continuous Bag-of-Words model
CNN	Convolutional Neural Network
CSMN	Context Sequence Memory Network
HOG	Histograms of Oriented Gradients
ILSVRC	ImageNet Large-Scale Visual Recognition Challenge
k-NN	K-Nearest Neighbors
LDA	Latent Dirichlet Allocation
LRCN	Long-term Recurrent Convolutional Networks
LSA	Latent Semantic Analysis
MSCOCO	Microsoft Common Objects in COntext
PCA	Principal Components Analysis
PMI	Pointwise Mutual Information
RNN	Recurrent Neural Network
SEMILAR	SEMantic simILARity toolkit
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features
Tf-Idf	Term Frequency–Inverse Document Frequency

1. INTRODUCTION

The use of the Internet instead of traditional written news sources has become increasingly popular in recent years. It is easier to reach to the millions of accessible news sources and there is an incredible increase in the size of the amount of text and visual data. It is not an easy task to manage, define and select important parts from the enormous data. Processing of this massive data of text and images of news is a comprehensive research area for image and video captioning and summarization of news articles.

Image captioning is an important application area that has been researched by the computer vision and natural language processing communities recently. Understanding image contents is a challenging problem and essential for image description. Automatic image captioning can be used in various areas such as generating captions for the images shared on social media, making life easier for blind people, helping journalists for creating captions for the news images and summarization of news articles.

Many studies have been published on image captioning. There are also news image caption generation works. Deep Neural Network architectures are built recently for Computer Vision and Natural Language Processing problems on image captioning and image retrieval for news.

Existing works on image caption generation focus on generating a new description for a given image. The problem of identifying image related sentences in news articles is examined in this study for the first time and our approach is different from the earlier works. We try to find the best description for the image, using the news article itself, by choosing the most appropriate sentences from the news text. Previous studies investigated creating subtitles related to an image by observing the objects and situations in that image. In contrast, in this study, the relationship between the image and the news is not broken as the most suitable sentences are selected from the news text itself. To do this, we use both image descriptors and text descriptors and try to

learn the relationship between them.

In this study, we target learning the relationship between the image and the most appropriate sentences in the news text. To learn this, we have created ground truth for a training set of news articles and their corresponding images. We represent the images using HOG (Histograms of Oriented Gradients) and BOVW (Bag of Visual Words) image features. Similarly, we represent sentences using the distributed vector representations of their words obtained using the Word2Vec model. The image-sentence pairs were categorised with Naive Bayesian, K-nearest neighbor and Random Forest classifiers and the success rates were examined.

The thesis is organized as follows: In Chapter 2, we review the literature about image captioning and the datasets used in this area. Chapter 3 describes the image and text representation techniques, as well as the classification methods. Chapter 4 describes our dataset, the methodologies we used to create the dataset and the ground truth generation techniques. Chapter 5 describes our proposed methodology and the experimental results. Finally, last words about this thesis and future ideas are given in Chapter 6.

2. RELATED WORK

Image captioning is an important application area that has been researched by the computer vision and natural language processing communities. With the increasing availability of images on the web, automatic image annotation has been an important and a challenging problem since the late 1990s as mentioned in [1]. To solve this problem, many different techniques have been offered by the research community. Cheng *et al.* [1] have a survey about automatic image annotation and they classify these automatic image annotation methods into five categories that are Generative model-based image annotation, Nearest neighbor-based image annotation, Discriminative model-based image annotation, Tag completion-based image annotation and Deep Learning-based image annotation.

Pan *et al.* [2] 's work which is one of the first works on image captioning, tries to find keywords as captions for the images by using the correlations between keywords and image features such as color, texture, shape, size, position. Image regions are labeled with a set of words called "blob-tokens" to construct a model. They propose four methods called Corr, Cos, SvdCorr, SvdCos to predict the words for the blobs. Corr method calculates the correlation with the co-occurence counts of the terms. Cos method calculates the cosine of the blob and word vectors. SvdCorr and SvdCos methods are the decomposed versions of the Corr and Cos methods.

Deschacht *et al.* [3] try to detect the salience and visualness of an entity mentioned in the text on the 100 image-text pairs of Yahoo! News dataset. These features helps to annotate the images by calculating the probability of that entity as being present in the accompanying image. The entities, visualness and salience give the best performance together.

Guo *et al.* [4] propose an automatic image annotation method that is an implementation of graph theory based on K-nearest neighbor graph. They use shape, spatial location, color and texture features of the images. The feature vectors of the images,

region and annotation words have adjacency with weighted edges on the graph.

Aker *et al.* [5], Fan *et al.* [6] and Plaza *et al.* [7] focus on captioning of GPS images. They only use the text features and summarization techniques by using the texts, documents and descriptions of the images. Aker *et al.* [5], use the summarizing technique of web documents that have information about the image location. Fan *et al.* [6], generate image captions by using the web content about an image. The generated image captions consist of the identified place name, a set of concept keywords, a set of expanded web keywords and a text paragraph of summary of the identified place. Plaza *et al.* [7], propose a graph-based and a statistically-based summarizing approach on web documents that describe the images of places.

Yu *et al.* [8] try to annotate images. They have an annotated images dataset. When an image is received by their system, they firstly segment image by using color intensity feature. After this process, they extract the objects of the image and match these objects with the training dataset to annotate the objects and images.

Krizhevsky *et al.* [9] use a deep convolutional neural network (CNN) to train and classify the 1.2 million images of the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC)-2010. ImageNet is a dataset of 15 million labeled images with 22.000 categories. The images are collected from the web and labeled by people working on Amazon's Mechanical Turk. This work classifies the dataset to 1000 different classes. Their large neural network has 650.000 neurons and five convolutional and three fully-connected layers. They declare that their work is better than the previous works on this dataset.

Mason [10] works on a domain-specific dataset of online shopping images and image captions that are product descriptions. They use a topic model to understand the image features and natural language descriptors. The model tries to estimate the likelihood of the words in the captions and generates a new caption for the query images.

Xi *et al.* [11] try to cluster the image regions with the weighted clustering algorithm. This technique helps to avoid to be dominated by the weakly relevant features by giving greater weights to the relevant features. This is calculated by the statistical distribution of the features. About the annotation, they calculate the conditional probability of the words and they find the relationship between the clustered image regions and the words of annotations.

Kılıçkaya *et al.* [12] try to find the most similar image to the query image and use the same caption of this similar image for the query image. The difference of this work is on the image representation part. They use the meta-class descriptor that is a high-level global image representation technique to understand the meaning of the image.

In the last years, Deep Convolutional Networks are mostly used to solve the image captioning problem. The works mentioned below are mostly using these techniques. Donahue *et al.* [13] propose Long-term Recurrent Convolutional Networks (LRCNs) architecture, with convolutional layers and long-range temporal recursion. Recurrent models are better than the previous works. In LRCN, the image features are extracted with the CNNs and these features are learned by sequence learning layers. By these layers the image captions are generated.

Fang *et al.* [14] propose a new model for image captioning. Their system learns about the adjectives, nouns and verbs of the image captions and which part of the image is corresponding to these words. They generate new captions for the query image by the model. After generating the captions, their deep multimodal similarity model re-ranks the captions and decides which one is the best. This work is the best on their time on Microsoft Common Objects in COntext (MSCOCO) [15, 16] dataset. This dataset contains captions generated by humans and the objects information in the images.

Chen *et al.* [17] propose the first bi-directional model for both generating a caption for the given image and generating visual features for the given caption by using

Recurrent Neural Networks (RNNs). They make experiments on numerous datasets such as the PASCAL sentence dataset [18], Flickr 8K [18], Flickr 30K [18], and the Microsoft COCO dataset [15, 16].

Karpathy *et al.* [19] propose a model generating natural language descriptions for the images and image parts by learning the inter-modal correspondences between images and image descriptions. For image regions, they use Convolutional Neural Networks and for sentences they use bi-directional Recurrent Neural Networks. After this step, they use a Multimodal Recurrent Neural Network to align the image regions and their descriptions. They use the Flickr 8K [18], Flickr 30K [18], and the Microsoft COCO dataset [15, 16] for the experiments.

Xu *et al.* [20] also works on the same datasets. They propose an attention based model to understand the meaning of the images by describing the content with object detection technique. They use standard backpropagation techniques and their model learns to find the objects in the image when it is trying to find the related words of the objects.

You *et al.* [21] propose a combining model of two techniques. The first one is the top-down technique that converts the images into words. The second one is the bottom-up technique that describes the regions or objects of the images. The proposed model learns to combine by using recurrent neural networks and learns how to attend the semantic proposals. This work also uses Flickr 30K [18], and the MSCOCO dataset [15, 16].

Fu *et al.* [22] propose a method using parallel structures of sentences and images. They generate the next word by controlling the previous ones and the image regions are compatible with the words. This is made by aligning the shifting on the image regions and the generated words.

Park *et al.* [23] try to generate the image descriptions such as hashtags and posts based on the personal data. They collected a dataset from Instagram and they have

the people's previous posts. By using the previous posts of the people, they estimate the new hashtag and post descriptions based on that user's style of posts. Their image captioning model is called Context Sequence Memory Network (CSMN).

Works on image captioning on different domains are mentioned so far. In this work, we are trying to identify the news text sentences which are related with the news images, so from here we will mention about the works about news image captioning.

Feng *et al.* [24, 25] propose extractive and abstractive models for caption generation. They have a probabilistic model that annotates the images by suggesting keywords for an image. After this suggestion, they rank the sentences to find the more informative one. The abstractive model results are higher than the extractive methods.

Jadhav *et al.* [26] propose a model that works on a dataset of news articles, images and captions. Firstly, they try to select content with using a probabilistic image annotation model by suggesting keywords. Their model takes the image captions and the articles as labels of images and summarizes the content according to the image.

Vijay *et al.* [27] propose a model that learns to create a caption for the images that are embedded with the related news articles by the stemming process with frequency ranking calculation for accurate caption generation.

Ramisa *et al.* [28] propose an adaptive Convolutional Neural Networks (CNN) architecture that shares most of the structure for multiple tasks including source detection, article illustration and geolocation of news articles. They introduce a novel dataset called BreakingNews. This dataset contains approximately 100K news articles with images, text and captions, and is enriched with heterogeneous meta-data such as GPS coordinates and user comments.

Hollink *et al.* [29] also propose another dataset for Deep Learning architectures on news articles. The dataset is called The ION and it contains 300K news articles that are published between August 2014 - 2015 in five online newspapers from two

countries. The dataset is formed as json files and it consists of publication date of the news, headline, article, caption of the image. The images themselves are not in the dataset but their feature vectors extracted by Convolutional Neural Network are included.

3. REPRESENTATION AND CLASSIFICATION METHODS

In this chapter, we review the feature extraction and classification methods. Firstly, we will mention about the techniques that help us to create the dataset. Afterwards, we will mention about the image representation techniques and text representation techniques respectively. Finally, we will mention about the classification methods.

3.1. Dataset Generation Methods

In this section, cosine similarity that is a distance calculation method, the text representation and similarity methods that are Term Frequency–Inverse Document Frequency (Tf-Idf), SEMILAR (SEMantic simILARity toolkit) and Word2Vec are analyzed.

3.1.1. Cosine similarity

Cosine similarity is a distance metric of the similarity between two vectors projected in a multi-dimensional space and measures the cosine of the angle between them. The vectors are similar at the maximum rate when they are parallel and if they are vertical, they are dissimilar. Cosine similarity is the dot product of the two vectors divided by the product of the two vectors' magnitudes. The formula of the cosine similarity is shown below at eq. (3.1), A_i and B_i are components of the A and B vectors.

$$similarity(A, B) = cos() = \frac{A \cdot B}{||A|| \times ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3.1}$$

Cosine similarity is commonly used to match similar documents. In this work, we use cosine similarity to compare two sentence vectors to find if they are similar or not.

3.1.2. Term Frequency–Inverse Document Frequency (Tf-Idf)

Term frequency–inverse document frequency (Tf-idf) [30–33] is a natural language processing method that is used for the research areas of summarization, information retrieval and text mining. This method is a numerical statistical measure to quantify what a document is about by indicating the importance of a word in a document.

Term frequency (tf) part stands for number of times a term occurs in a document, helps to measure the frequency of a term. Term frequency is first mentioned by Luhn [34]. Luhn [34] says that the more frequent terms are considered more valuable in proportion to their observed frequencies. Document lengths differ and it is possible that a term appears much more times in long documents than shorter documents. Hence, the term frequency is often divided by the document length for normalization using the formula below:

TF(w) = (Number of times term w appears in a document) / (Total number of terms in the document)

Inverse document frequency (Idf) part measures how important a term is. Idf reduces the weight of commonly used words and increases the weight of the words that are used rarely in documents. For example, while computing Tf, all term importances are considered as equal. However it is known that commonly used words, such as ”is”, ”a”, and ”that”, may appear a lot of times but have little importance. Therefore, we need to weigh down frequently used terms when scaling up the rare ones. Inverse document frequency is first mentioned by Jones [35]. Jones [35] says that the specificity of a term can be quantified as an inverse function of the number of documents in which it occurs. Idf is computed using the formula below:

IDF(w) = log_e(Total number of documents / Number of documents with term w in it)

An example of tf-idf calculation is given in [36]: If the word "*apple*" appears 3 times in a document of 100 words, the tf for "*apple*" is then (3 / 100) = 0.03. If there are 10 million documents and the word "*apple*" appears in one thousand of these, then, the idf is calculated as log(10,000,000 / 1,000) = 4. Thus, the Tf-idf weight is the product of these amounts: 0.03 * 4 = 0.12.

In this work, we use a simple version of tf-idf for comparing two sentences similarities by counting the occurrences of every word in both sentences. The counts of every word construct a vector and we calculate the cosine similarity. The threshold value is determined as 0.35. The details of the use of this method is mentioned in Chapter 4 of this work.

3.1.3. SEMILAR (SEMantic simILARity toolkit)

SEMILAR [37] is a semantic similarity toolkit which uses various known word-to-word and sentence similarity measures and models. SEMILAR offers a variety of similarity methods based on WordNet, Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA), BLEU, Meteor, Pointwise Mutual Information (PMI), methods that use syntactic dependencies, optimized lexico-syntactic matching methods based on Quadratic Assignment and methods that incorporate negation handling [38].

SEMILAR contains methods to measure the semantic similarity by word-to-word measures, sentence-to-sentence measures, paragraph-to-paragraph measures and document-to-document measures [38]. In this work, we need sentence-to-sentence similarity. Among the sentence-to-sentence similarity methods, we selected the WordNet [39] Lesk Tanim method.

WordNet [39] is a grand lexical database of English. Nouns, adjectives, verbs and adverbs are organized semantically. Synonymous words are grouped into cognitive syn-

onym sets (synsets) [40]. Each synset expresses a different concept. Synsets are linked through conceptual-semantic and lexical relationships. WordNet is a combination of dictionary and thesaurus. A word is polysemous if it is founded in several synsets, where each synset is set up to represent a possible word meaning. For example "*base*" word occurs in two noun synsets, (*base, alkali*) and (*basis, base, foundation, fundament, groundwork, cornerstone*), and the verb synset (*establish, base, ground, found*).

Wordnet Lesk Tanim [41] is a semantic association algorithm that is based on and uses the Wordnet database. The degree of similarity between the two words is determined by calculating the number of overlaps in their dictionary definitions. The original Lesk algorithm [42] concretizes words in short phrases. The definition of each meaning of a word in a sentence is compared to the definitions of each other in the sentence. The definition of a word is the perception that shares the most words in common with the definitions of other words. The original Lesk algorithm is based on definitions found in traditional dictionaries. Pederson *et al.* [41] evolve Lesk's basic approach. This new approach benefits from interrelated relationships between the synonyms that WordNet offers and is able to compare the definitions of words related to words to be clarified. A vector containing the co-occurrence counts of words is constructed. Relatedness is determined by comparing this definition vector using the cosine similarity measure.

In this work, SEMILAR API [38] is used as a jar library to compare sentence-to-sentence similarity. There are two types of Lesk algorithm, the first one is the greedy Lesk Tanim, the second one is the optimal algorithm. The greedy algorithm checks each word with the Wordnet database and does not use previously acquired senses. The optimal algorithms combined with heuristic techniques give more accurate results by eliminating the not matching definitions. We used the optimal Lesk Tanim algorithm. Before calculating the similarity between strings, the strings are pre-processed. The role of preprocessing is to obtain a variety of grammar and linguistics information, such as parts of speech, lemmas and syntactic dependencies, that are very useful in the calculation of semantic similarities between the two texts. There are four preprocessing steps available which are tokenization, extraction of word's base forms, part-of-speech

tagging and syntactic parsing. Tokenization is the separation of punctuation marks from words. The extraction of the basic forms of the word can be obtained as lemmas or stems. We use the stemming version in this work.

3.1.4. Word2Vec

Word2Vec [43–45] is an unsupervised (no labels) and prediction-based model that tries to express words in vector space. Word2Vec is a two-layer neural network that handles text.

Word Embedding is text converted to numbers and may have different numeric representations of the same text. It is vector representation of a particular word. For example, if we look to the sentence "This sentence is an example sentence". A word in this sentence may be "this" or "example". The dictionary can be a list of all unique words within the sentence. So, the dictionary will be like ['This', 'sentence', 'is', 'an', 'example']. A vector representation of a word is a one-hot encoded vector. In this vector illustration, 1 stands for the position where the word exists and 0 everywhere else. The vector representation of "example" according to the above dictionary in this format is [0,0,0,0,1] and of "this" is [1,0,0,0,0]. If we visualize these encodings, we can think of a 5 dimensional space, where each word occupies one of the dimensions and has nothing to do with the rest. This is a very simple method to show a word in vector format. The words in the same context must be close to spatial positions to get better representations.

The purpose and usefulness of Word2Vec is to group the vector of similar words together in vector space by mathematically identifying similarities. Word2Vec creates vectors that are distributed as digital representations of word features, such as the context of the individual word properties. If the context of two words is similar, they will have similar vector representations.

There are two main Word2Vec architectures used to produce a representation of words: Continuous Bag-of-Words model (CBOW) and the Skip-Gram model. Algorith-

mically, these models are similar and both of these are shallow neural networks which map words to the target variable which is also a word [46]. Both of these techniques learn the weights that act as word vector representations. Continuous Bag-of-Words model (CBOW) predicts the current word from a window of context words surrounding it. Skip-Gram model uses the existing word to estimate the window surrounding the context words. For example, while CBOW estimates target words (e.g. 'table') from source context words ('the book is on the'), skip-grams do the opposite and estimate source context words from target words. CBOW is faster while skip-gram is slower but does a better job for rare words. Figure 3.1 shows the difference of methods.

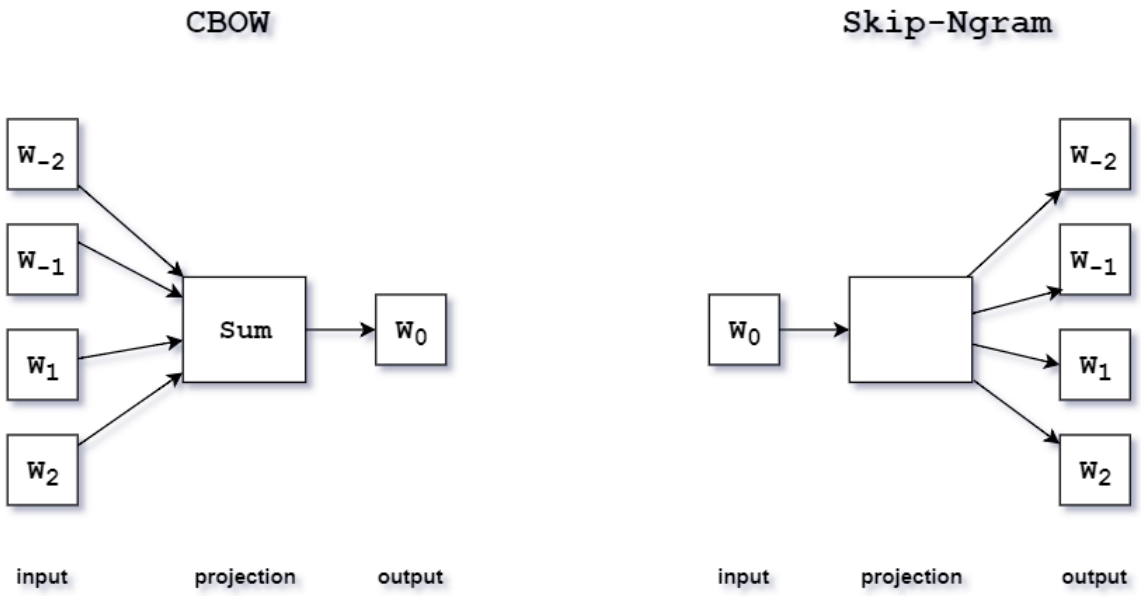

Figure 3.1. CBOW and Skip-Gram Models

Usually, Word2Vec is trained with Skip-Gram model and in this work, the Skip-Gram architecture is used and we will explain about this model. Skip-Gram architecture's aim is mentioned above. It tries to define the context as the window of the words to the right and the left of a target word. One of the most important hyperparameter for Word2Vec is window size. Figure 3.2 shows an example of windows size 2 for the sentence "The quick brown fox jumps over the lazy dog." [47].

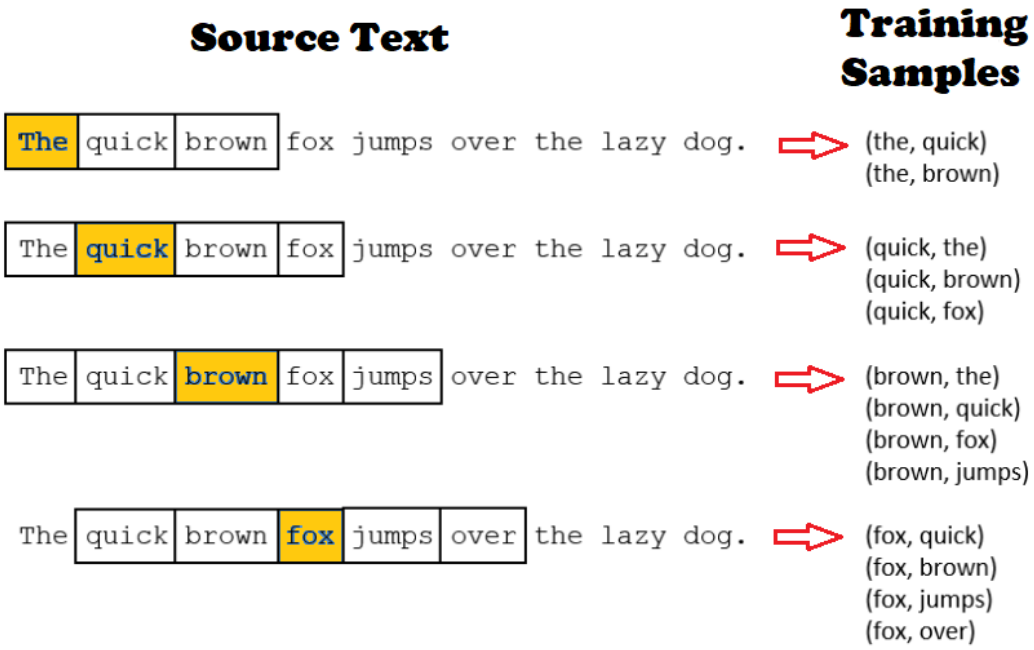

Figure 3.2. Window size sample

Considering the Figure 3.2, we see context and target pairs. Skip-gram model tries to predict each context word from its target word, so according to our example, it tries to predict 'the', 'brown' and 'fox' from 'quick' and 'the', 'quick', 'fox' and 'jumps' from 'brown'. The dataset is constructed from these input and output pairs. The network learns the statistics from the number of times each pairing exists. So, for example, the network will probably take more training examples of certain semantic relationships such as male-female, verb tense and country-capital relationships between words. For example, if you give it the word "Turkey" as input, then it will output a much higher probability for "Ankara". Another sample can be the basic algebra presentation between the vectors of "king - man + woman = queen" that gives the feeling of word relationships. An example of word relationships is shown in Figure 3.3 [46].

The architecture of neural network of Word2Vec is shown in Figure 3.4 [47]. If we have a vocabulary of 10.000 unique words, we will give an input word like "apple" as a one-hot vector and the output of the network will be a single vector containing the

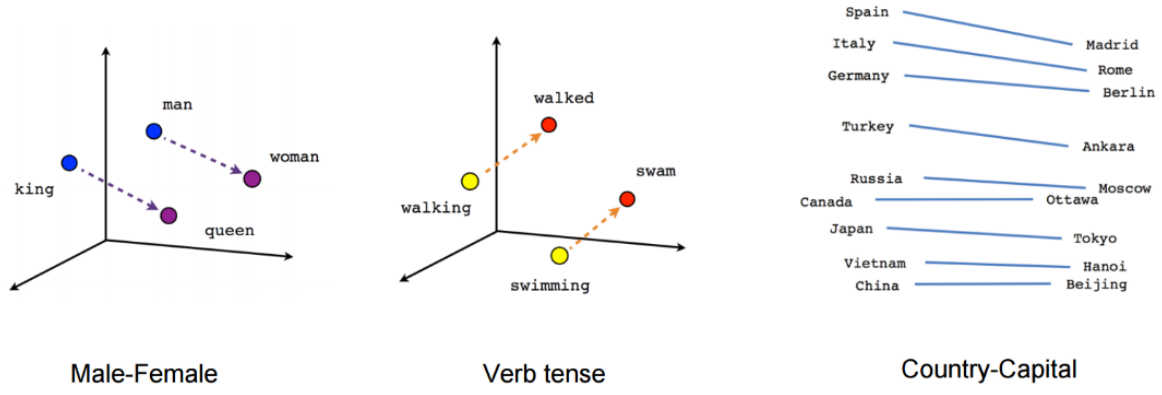

Figure 3.3. Visualization of the relationships between words [46]

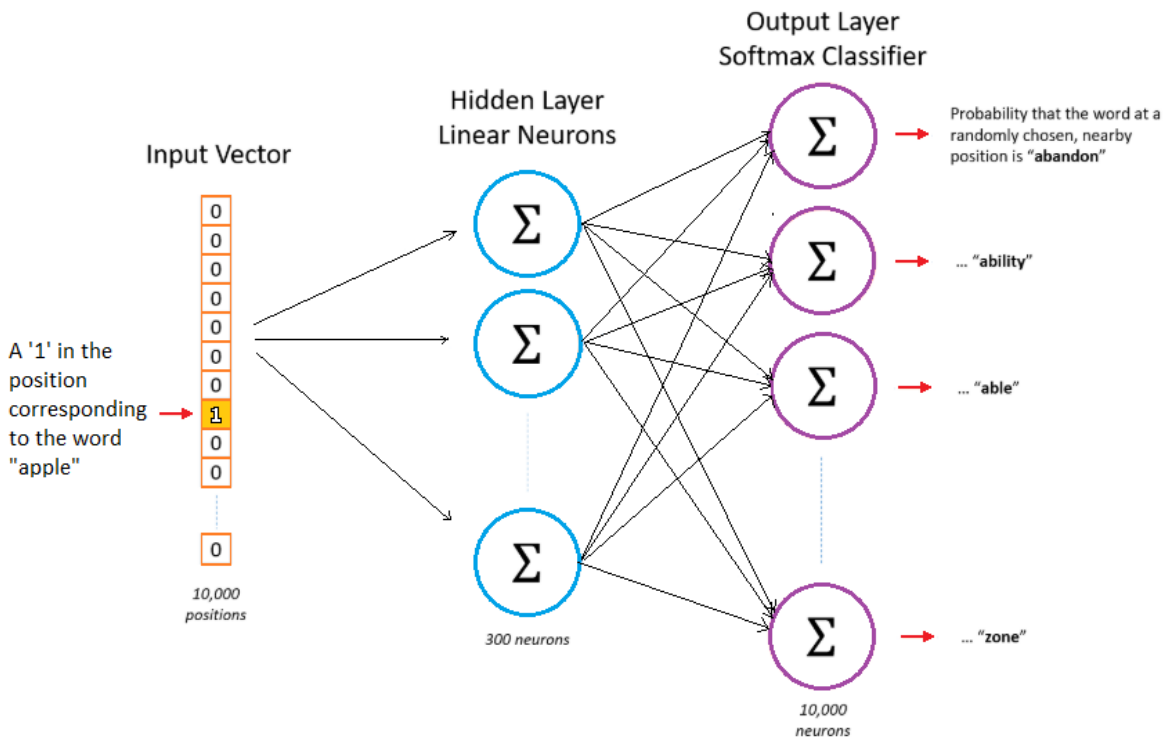

Figure 3.4. Application of softmax between hidden layer and output layer consisting of 10000 unique words and calculation of probability

probability for every word in our vocabulary, that a randomly selected nearby word is that vocabulary word. The hidden layer has no activation function on the neurons, but the output neurons use softmax. If we need word vectors with 300 features, the hidden layer is going to be represented by a weight matrix containing 10.000 rows (one for every word in our vocabulary) and 300 columns (one for every hidden neuron). The number of nodes in the hidden layer tells us how many words are represented in space and is another hyperparameter for the Word2Vec model like window size hyperparameter.

3.2. Image Representation Methods

In this section, image representation techniques of our work are described. Histograms of Oriented Gradients (HOG) and Bag of Visual Words (BOVW) methods are used for image representation and their details are given on this section, respectively.

3.2.1. Histograms of Oriented Gradients (HOG)

HOG is a feature descriptor used to detect objects in computer vision. Dalal and Triggs [48] used HOG in their study to detect pedestrians from static images. A feature descriptor is a representation of an image or an image patch that turns the image information into numbers by extracting useful information and discarding irrelevant information.

HOG descriptor technique uses the histogram of frequencies of gradient orientation presences in local parts of an image. Gradients (x and y derivatives) of an image are useful because the magnitude of gradients is large around corners and edges which are the regions of rapid intensity changes. Accordingly, edges and corners give a lot of information about an object than flat regions. The image is divided into image patches called cells with determined sizes, and the histogram of gradient directions is calculated for the pixels within each cell.

The first step is calculation of the horizontal and vertical gradient values. Figure 3.5 show these horizontal and vertical filter kernels. Each pixel has the size and

direction of the gradient. The images that have three channels of color are evaluated with these three channels and the magnitude of the gradient is the maximum of the magnitude of the gradients of three of them. The angle is the angle corresponding to the maximum gradient.

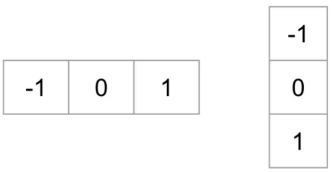

Figure 3.5. Gradient filters

The second step is creating the cell histograms by dividing the image. The histogram contains 9 bins corresponding to angles 0, 20, 40, 60, 80, 100, 120, 140, 160. The descriptor is the concatenation of these histograms. Figure 3.6 shows the details about histogram creation process. The green circled pixel has 80 degrees angle and its magnitude is 2. So this magnitude is moved to the 5th bin. The red circled pixel has 10 degrees direction and its magnitude is 4. 10 degrees is not selected as one the 9 bins angles so, its value is divided between 0 and 20. Its vote for the bin is moved to 1st and 2nd bins as 2 and 2.

The next step is normalizing the histogram descriptor to be independent of lighting variations. This process is accomplished by grouping the cells together into larger, spatially bound blocks. The HOG descriptor is the concatenated vector of the ingredients of the normalized cell histograms from all of the regions of blocks. These blocks typically overlap, that is, each cell contributes more than once to the final descriptor. Figure 3.7 shows that, HOG feature vector is created by concatenation of the histograms of blocks and cells.

3.2.2. Bag of Visual Words (BOVW)

Bag of Visual Words (BOVW) [49] is an image classification and image retrieval technique in computer vision that uses the small regions of a picture as words. This

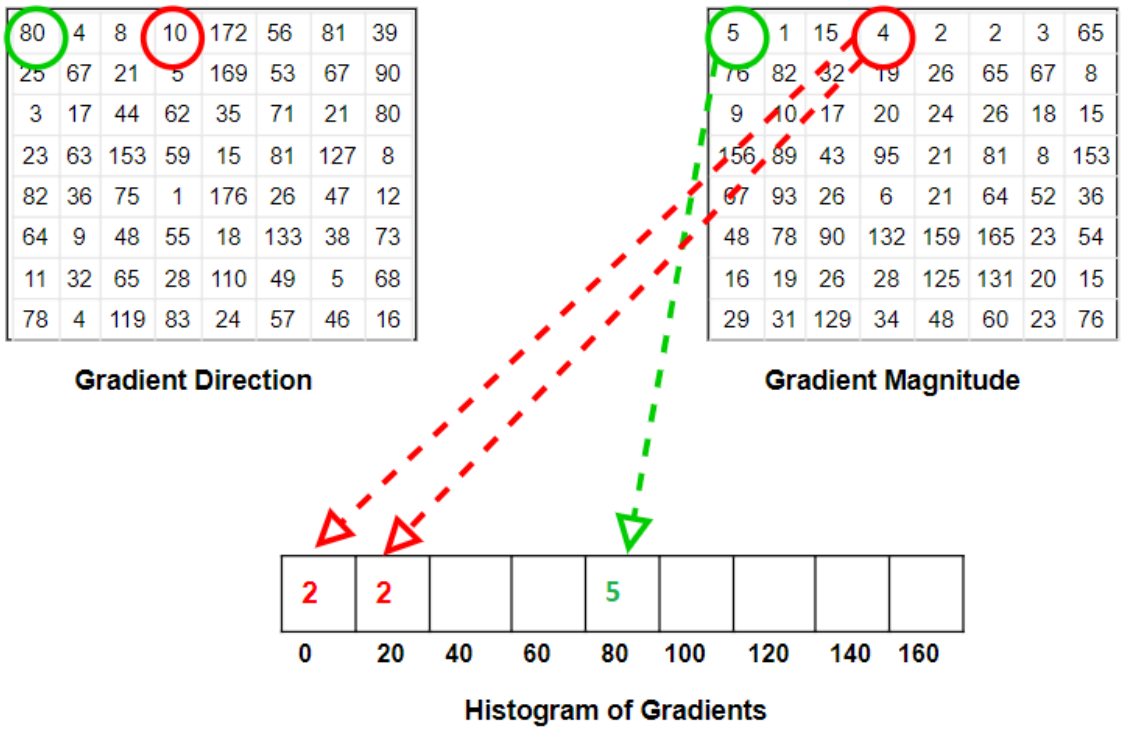

Figure 3.6. Histogram of gradients

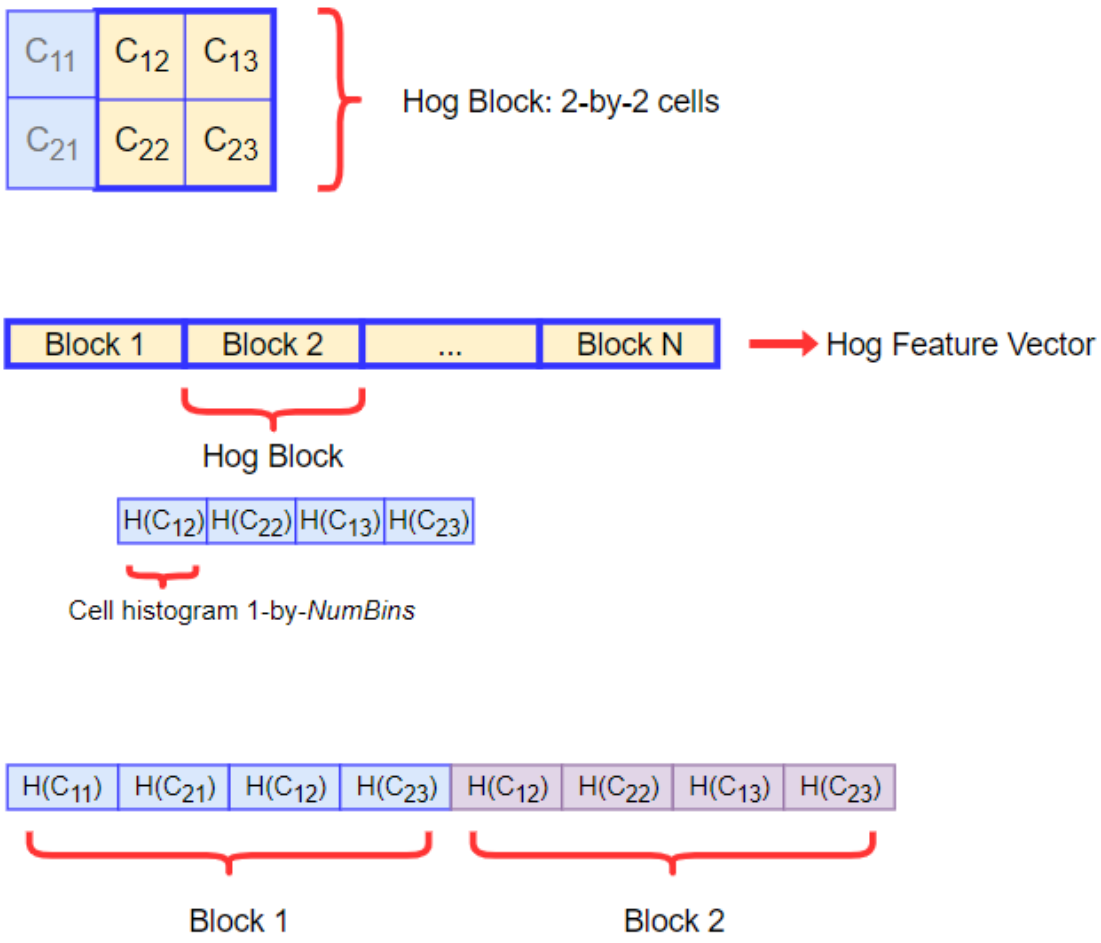

Figure 3.7. HOG blocks and HOG feature vector creation

approach is similar to the Bag of Words technique that is used for the words in the text of a document. BOW counts the words of the document and measures the frequencies of the words. These counts are the frequency histograms and describes the document as a vector. BOVW creates a vector representation of an image by counting words in the image, that is the histogram of words. These image words are the image features. The words construct the vocabulary from all the image patches. Figure 3.8 shows the main idea of BOVW [50].

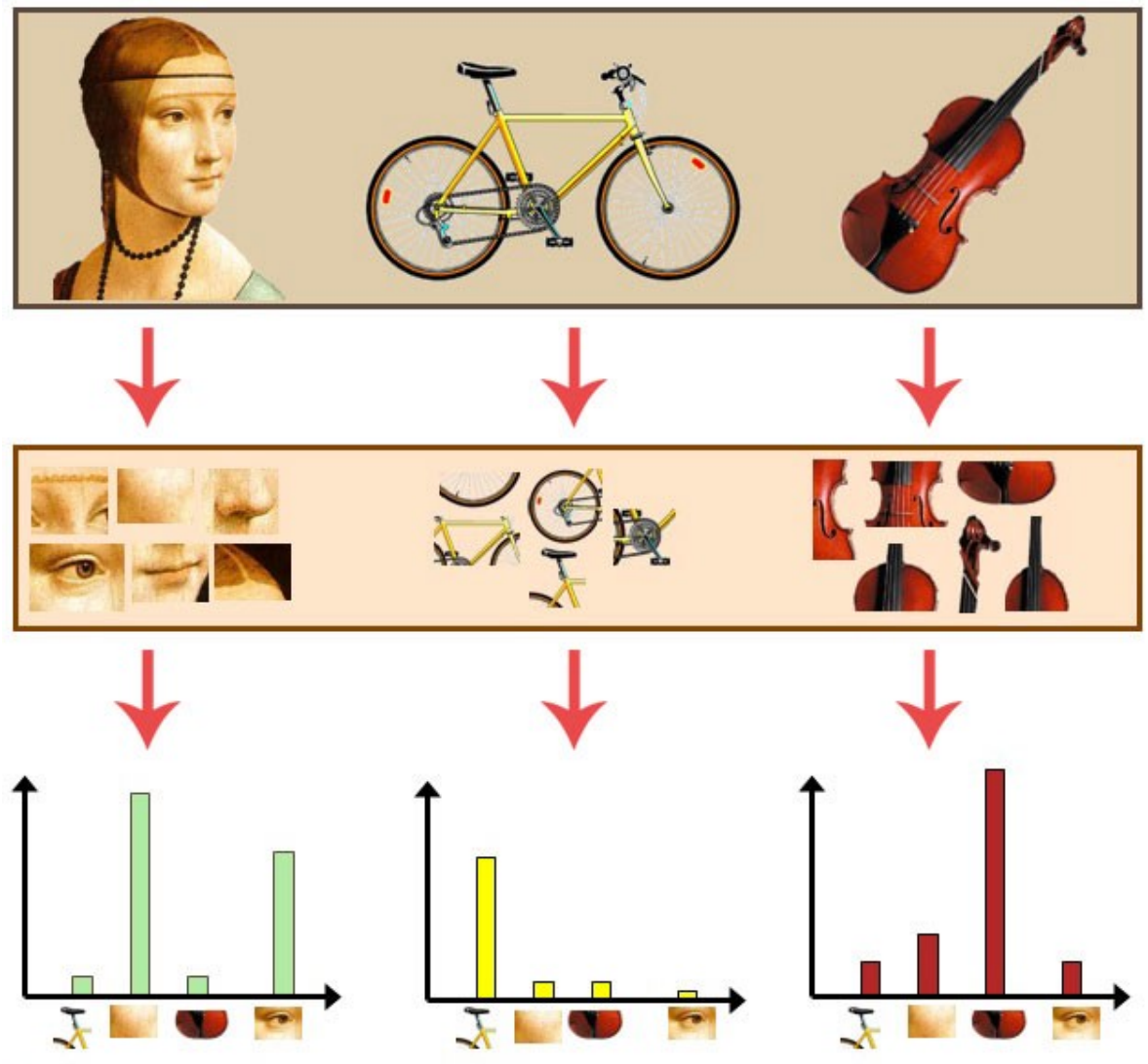

Figure 3.8. Histogram of visual words [50]

The main idea of BOVW is to represent the images as feature sets that are the keypoints and descriptors of the images, this is called bag of features. The images that

share large number of features or keywords are related to each other. Therefore, the first step in building a BOVW is to extract features by getting descriptors from each image.

For creating the vocabulary, we need to find the features of the images. A feature is any noticeable and important point or point group in the image. These image properties are used to measure the similarity between images and can be accomplished in different ways. These ways may include general image features such as color, texture, shape, corner points, edges, blobs or may be local image features such as Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF) or Histogram Of Gradients (HOG) depending on the application. The methods of showing these features relate to how to show the patches as vectors. Transforming to numerical vectors is shown in Figure 3.9. These vectors are called feature descriptors. A good descriptor should have the ability to cope with some degree of intensity, rotation and scaling variations. So, we use Speeded-Up Robust Features (SURF) [51] method to find the important local keypoints in the images of our dataset.

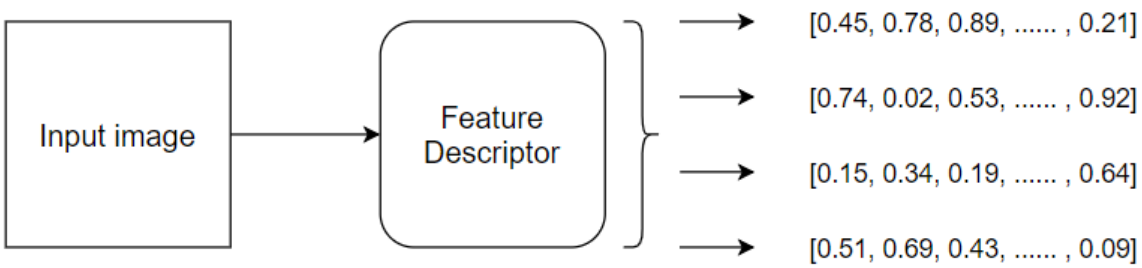

Figure 3.9. Image features to numerical vectors

This method is a scale and rotation invariant interest point detector and descriptor like Scale Invariant Feature Transform (SIFT), but the speeded-up version of it. It can be used to recognize objects and people and image classification. SURF uses box filters as an approximation of Laplacian of Gaussian smoothing in each location in the image under different scales. It can be done parallel for these different scales. The feature detector is durable because it applies masks along each axis and applies

masks at 45 degrees to the axis. To find points of interest, SURF uses a blob detector based on the Hessian matrix. Hessian matrix box filter is shown in Figure 3.10 [52] and the Hessian matrix is shown in equation 3.2. L_{xx} is the convolution of the second derivative of a Gaussian with the image at the point.

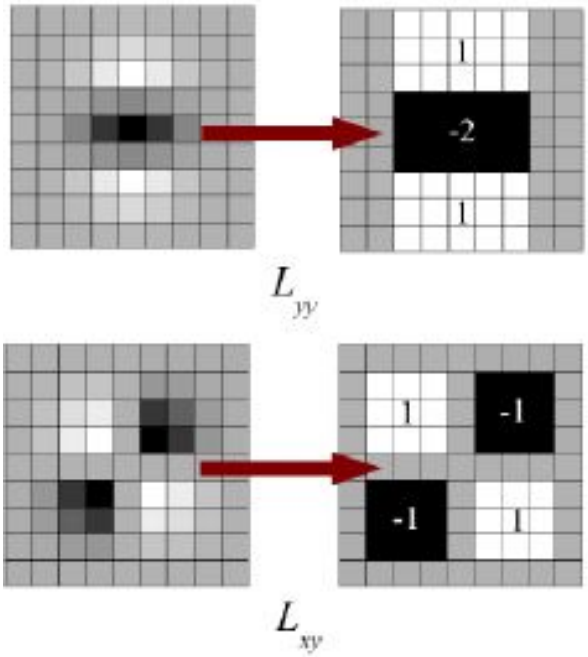

Figure 3.10. Hessian box filter

$$H = \begin{vmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{vmatrix} \tag{3.2}$$

If the local change which is measured by the determinant of the Hessian matrix around a point, is maximal, that point is selected. Scale spaces are implemented by applying box filters of different sizes. Sums and responses in horizontal and vertical direction of Haar wavelet components are used to find keypoints. Neighbourhood around the keypoint is taken. It is divided into subregions. For each subregion, a vector is formed from the horizontal and vertical wavelet responses. Rotation is accomplished by finding the dominant direction of the property and rotating the sampling window

in line with this angle. When the returned neighborhood is obtained, it is divided into 16 sub frames, each sub frame divided into 4 squares [53]. For each sub-region, horizontal and vertical wavelet responses are taken and a vector like this is created, $v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$ [52]. This gives a vector of SURF feature descriptor with 64 dimensions. SURF feature descriptor extends 64 dimension to 128 dimension. The sums of d_x and $|d_x|$ are computed separately for $d_y < 0$ and $d_y \geq 0$ and vice versa for d_y. After this step, each image is a collection of 128 sized vectors. The selected keypoint descriptors and vectors are shown in Figure 3.11.

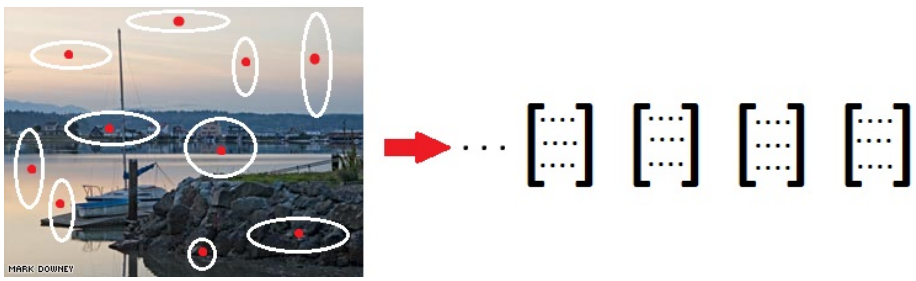

Figure 3.11. Detecting features and extracting descriptor

SURF converts each patch to 128-dimensional vector. After this step, each image is a collection of 128 sized vectors. After extracting the feature descriptors, we should create the bag of BOVW model. The feature vectors that stands for the patches of the images are clustered by k-means clustering algorithm and the centers of the clusters represent the words of the vocabulary. Descriptors clustering is shown in Figure 3.12.

k-means clustering aims to divide the observations into sets of k, each serving as a prototype of clusters, where each observation belongs to the nearest mean to the cluster [54]. Each data is a n-dimensional real vector, including a $x_1, x_2, \ldots, x_n$ dataset and K is given as the number of clusters. K-means clustering aims to divide the n data into a set of S = $S_1, S_2, \ldots, S_k$ to minimize sum of squares. According to the operating mechanism of the k-means algorithm, k objects are randomly selected to represent the center point of each set. The remaining objects are included in the clusters where they are most similar, taking into account their distance from the mean values of the clusters. Then, by calculating the average value of each set, new cluster

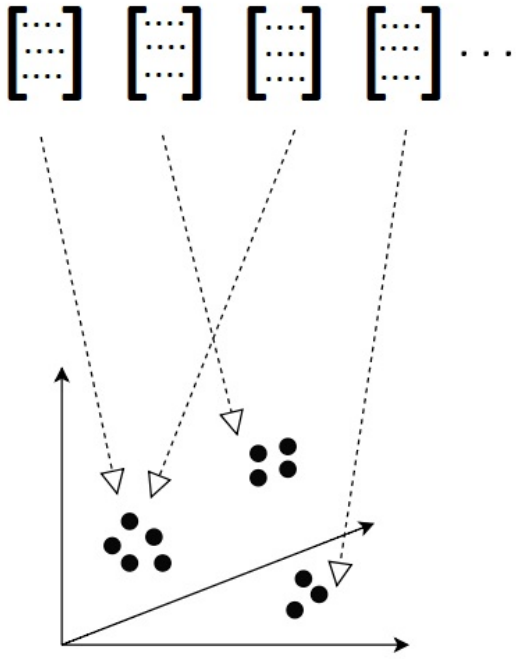

Figure 3.12. Descriptors clustering

centers are determined and the distances of objects to the center are examined again. The algorithm continues to repeat until there is no change.

In our BOVW model, k determines the number of words in the vocabulary. At the last step, the counts of the words in an image forms the histogram vector of words that represents the image. We used 500, 1000 and 2000 sized vocabulary and histogram vector on this work for comparison.

3.2.3. Principal Components Analysis (PCA)

The Principal Components Analysis (PCA) is used as a statistical method in image processing areas such as image compression and pattern recognition. It allows the datasets that have feature vectors that consist of related high-dimensional to be expressed with attribute vectors called a less-sized "principal component" without losing their properties.

The steps of the PCA method are shown in Figure 3.13.

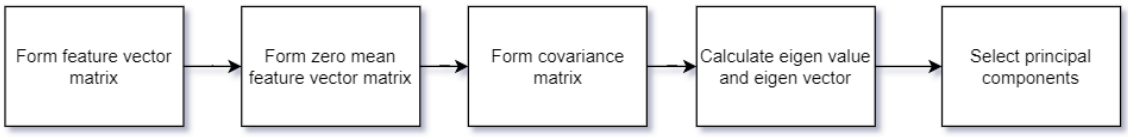

Figure 3.13. Principal Components Analysis (PCA) method steps.

The set of feature vectors of the dataset is converted into the M * N dimension X matrix, with the M number of samples and the N number of vector attributes (3.3).

$$X = \begin{bmatrix} x_1^1 & x_1^2 & ... & x_1^M \\ x_2^1 & x_2^2 & ... & x_2^M \\ ... & ... & ... & ... \\ x_N^1 & x_N^2 & ... & x_N^M \end{bmatrix} \tag{3.3}$$

To form a covariance matrix, a zero mean centered feature vector matrix must be created. To do this, first averaging the average vector values of feature vectors is taken as the average vector in μ (3.4) is calculated as shown.

$$\mu = \frac{1}{M} \sum_{i=1}^{M} X^i = \begin{bmatrix} \mu_1 \\ \mu_2 \\ ... \\ \mu_N \end{bmatrix} \tag{3.4}$$

The calculated average vector is substracted from each sample in the X matrix to obtain a zero mean centered feature vector matrix ($\widehat{X}$) (3.5).

$$\widehat{X} = \begin{bmatrix} x_1^1 - \mu_1 & x_1^2 - \mu_1 & ... & x_1^M - \mu_1 \\ x_2^1 - \mu_2 & x_2^2 - \mu_2 & ... & x_2^M - \mu_2 \\ ... & ... & ... & ... \\ x_N^1 - \mu_N & x_N^2 - \mu_N & ... & x_N^M - \mu_N \end{bmatrix} \tag{3.5}$$

The zero mean centered feature matrix is multiplied by the transpose matrix, and the C covariance matrix is calculated as the equation (3.6).

$$C = \widehat{X}\widehat{X}^T \tag{3.6}$$

The eigenvalues of the C covariance matrix and the eigenvectors corresponding to these eigenvalues are calculated by the equation in (3.7).

$$Cv = \lambda v \tag{3.7}$$

In (3.7), λ is the eigenvalue and v is the eigenvector.

Calculated eigenvalues show how well the eigenvector represents the samples. For this reason, the eigenvalues are sorted from the largest to the smallest, using the eigenvectors corresponding to the largest P eigenvalue, and the columns are calculated

by the equation in the W projection matrix (3.8), which performs the best projection of these eigenvectors.

$$W = \begin{bmatrix} w_1 & w_2 & ... & w_P \end{bmatrix} \tag{3.8}$$

The features of the dataset are obtained by projecting the W projection matrix on the space determined by the eigenvectors and a reduced y matrix is obtained (3.9).

$$y^i = W^T x^i \tag{3.9}$$

3.3. Text Representation Methods

In this section, text representation technique of our work is declared.

3.3.1. Word2Vec

Word2Vec method is used for text representation, the same technique as one of the Dataset Generation Methods in Section 3.1.4. The method details are described in Section 3.1.4 for Word2Vec.

3.4. Classification Methods

In this section, classification techniques of our work are declared. Naive Bayes, k-Nearest Neighbours and Random Forests methods are used for classification and their details are given on this section, respectively.

3.4.1. Naive Bayes

Naive Bayes is a classification method based on Bayes' theorem with an assumption of independence between features. The Naive Bayes classifier aims to determine the class of the samples in the dataset according to the probability principles set by Bayes' theorem.

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \tag{3.10}$$

In equation 3.10, X = (x_1, x_2,..., x_n) are the samples from our dataset, C = (C_1,C_2,...,C_m) are the classes of the dataset. P(C_i—X) is the posterior probability of class given predictor. P(C_i) is the prior probability of class. P(X—C_i) is the likelihood which is the probability of predictor given class. P(X) is the prior probability of predictor [55]. Naive Bayes classifier assumes that the effect of the value of a predictor (X) on a given class (C_i) is independent of the values of other predictors. The value of the probability of X being in C_i class is P(C_i—X). This value is calculated for all classes and the class with the maximum value is defined as the class of X sample.

3.4.2. K-Nearest Neighbours (k-NN)

K-nearest neighboring classifier is based on the principle of determining the class of a sample from the test data where that sample has the most samples within the neighboring instance of training dataset.

The classification success may vary according to the k parameter. As shown in the Figure 3.14, the class of the test data indicated by the green circle is red when given k = 3 and blue when given as k = 5.

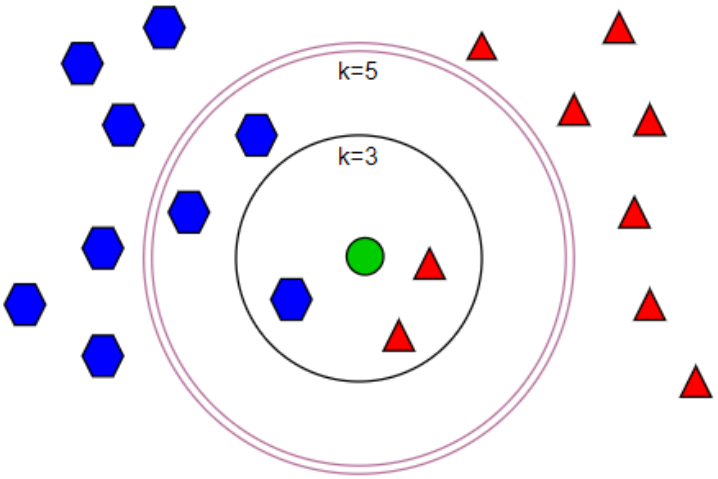

Figure 3.14. k = 3 and k = 5 for k-Nearest Neighbor classification example.

In this method, distance functions between neighbors are calculated and distance functions are used to determine the nearest k neighbor. These distance functions are shown in (3.11), (3.12) and (3.13).

$$L_1(x,y) = \sum_{i=1}^{n} |x_i - y_i| \tag{3.11}$$

$$L_2(x,y) = (\sum_{i=1}^{n} |x_i - y_i|^2)^2 \tag{3.12}$$

$$cos(\theta) = \frac{XY}{|X||Y|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \tag{3.13}$$

In this study, L_2 distance function is used in equation 3.12.

3.4.3. Random Forests

Random Forest classifier uses multiple decision trees in the training phase. For the training of these decision trees, random sets of sub-data sets are generated from the original training dataset. 2/3 of these subsets are used to create the decision tree structure while the other part is used to test the tree structure.

The random forest classifier uses the Gini index, which shows the homogeneity of the classes in the selection of attributes for branching in decision tree formation. For a given node t, the Gini index is expressed by the equation (3.14).

$$Gini(t) = 1 - (\sum_{i=1}^{n} P(i|t)^2) \tag{3.14}$$

In the equation (3.14), $P(i|t)$ represents the relative probability for class i in node t.

If the Gini index of a child node is lower than the Gini index of a parent node, this indicates that the branch is successful and that the node is close to homogeneity.

The number of trees to be created for this classifier and the number of samples in each node in these trees are determined by the user. When creating tree structures for randomly created subset sets of data, a random m variable is selected from within these dataset sets, and the variable that divides the branch in the best way is determined from these variables. The process of allocating nodes continues until it reaches the leaf node.

When classifying the new dataset with this classifier, the dataset is passed through all the tree structures created and receives votes from each class and assigns the class with the most votes as the class of the dataset.

In the Figure 3.15, an example of random forest classification process is shown. The test sample whose class is unknown is classified from all decision tree estimates by assigning it to the class with the highest probability value over the sum of the probability values taken for each class.

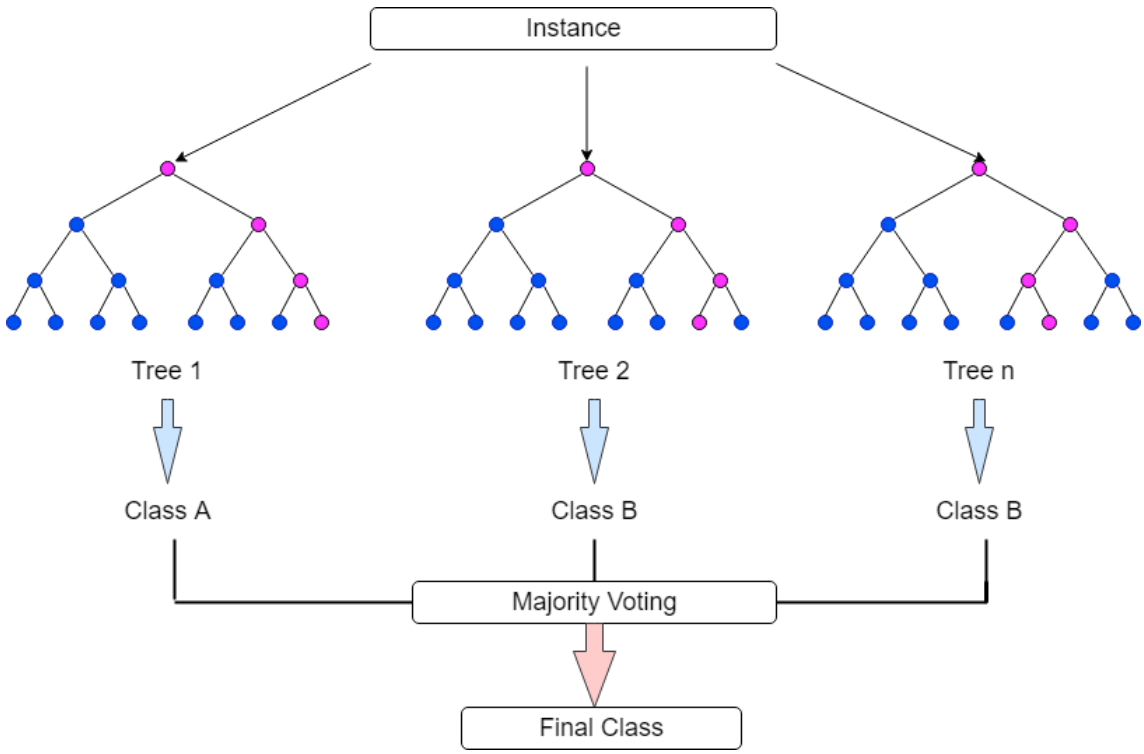

Figure 3.15. An example of random forest classification.

4. DATASET

In this chapter, we describe the preparation of the dataset used in this work.

In this work, our aim is to find the related sentences with the image of a news article. To achieve this goal, we need a dataset of news articles. When we searched about it, we found the work of Narayan *et al.* [56]. They developped a general framework for single document summarization with side information. To train their model, they used an augmented version of the CNN news highlights dataset [57]. Hermann *et al.* [57] crawled 93K CNN articles to build a large-scale corpus to set a benchmark for deep learning methods and the dataset has been used for single-document summarization. Narayan *et al.* [56] augmented this dataset with side information to see the effect of side information of news articles for summarization. They extracted titles and image captions of news articles if images exist, and they associated them with the corresponding articles. They used this augmented dataset for summarization.

Since our aim is to find the related sentences of news images, we also need the news images of the news articles. Therefore, we used the modified version of Narayan *et al.* [56]. They use the dataset for summarization of news articles, we used the inner text and image captions parts of the articles of the dataset for image caption identifying. The dataset just includes the captions of news images but we need the images themselves. For this purpose, we used a script for collecting the images of the news articles. All the dataset has 83.564 training and 1091 test news articles. Not all the news contains an image caption, 13.302 of the training set and 565 of the test set have image captions. We could not use of all these data for several reasons. One of these reasons is that, while collecting the images of the news, we could not reach some of the images because they were deleted from the web site. Another reason is, some articles have video captions and these video captions are not meaningful sentences, but just information about the video duration. An example of this kind of image and image caption is shown in Figure 4.1. As seen in Figure 4.1 while a normal image has a caption describing the image, a video just has a caption without detail and with video

duration and it is not an image. Accordingly, we eliminated these kinds of images and image captions.

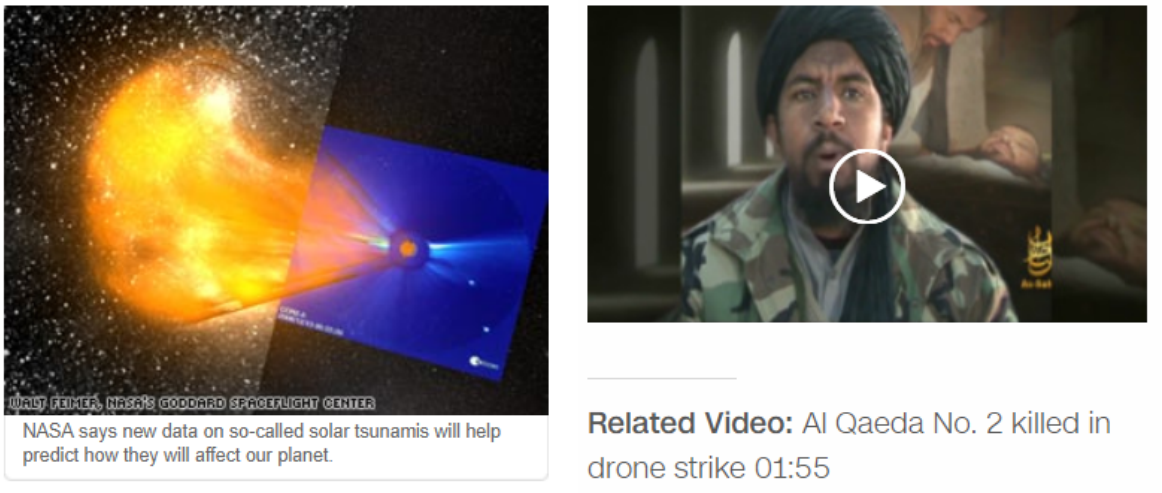

Figure 4.1. A normal image caption and a caption with video duration

The last reason is that some of the news have multiple images and image captions. If the news article has one image, the image caption really describes the image itself. But when the number of images increases, the image captions tell another story that contribute to the news main text, and this does not help us for finding related sentences of the image, therefore we eliminated the news that has more than one image. An example of a news which has more than one image is shown in Figure 4.2. As seen in the figure, the image captions are telling the news itself.

At last, we gathered 5.755 unique news images for training and 70 unique news images for test. The news images are in jpg format and most of the news images have 292x219 size. Figure 4.3 demonstrates a view from the crawled image dataset. The training dataset has 152.183 and test dataset has 1.283 sentences.

To identify the image related sentences from the news article text, we need to label the sentences as being related to the image or not. As we mentioned above, if a news article contains only one image, the caption of the image is related to the image. Therefore, we tried to find the image related sentences by using the caption. If

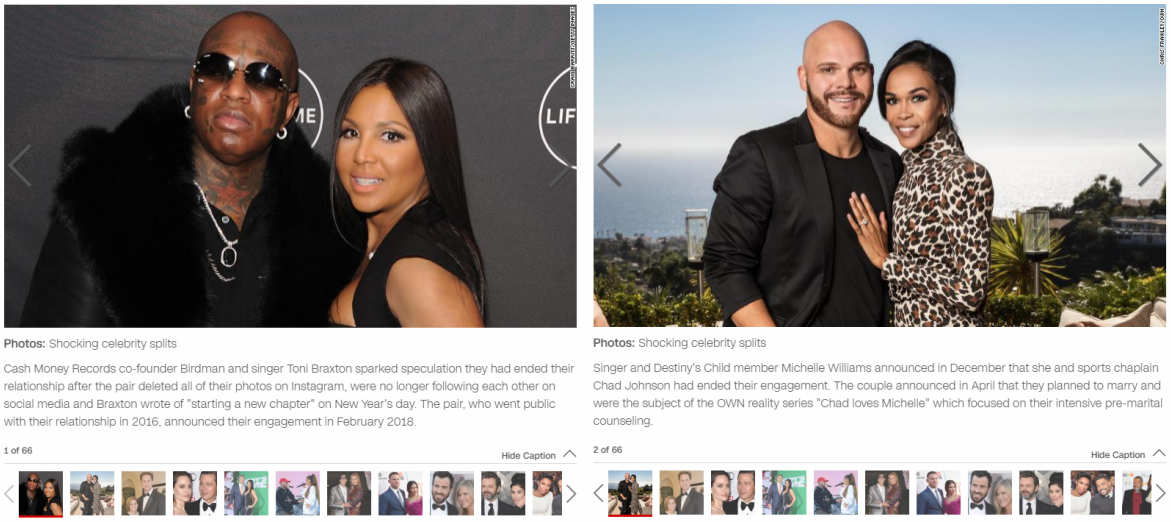

Figure 4.2. Example images of a news with more than one image

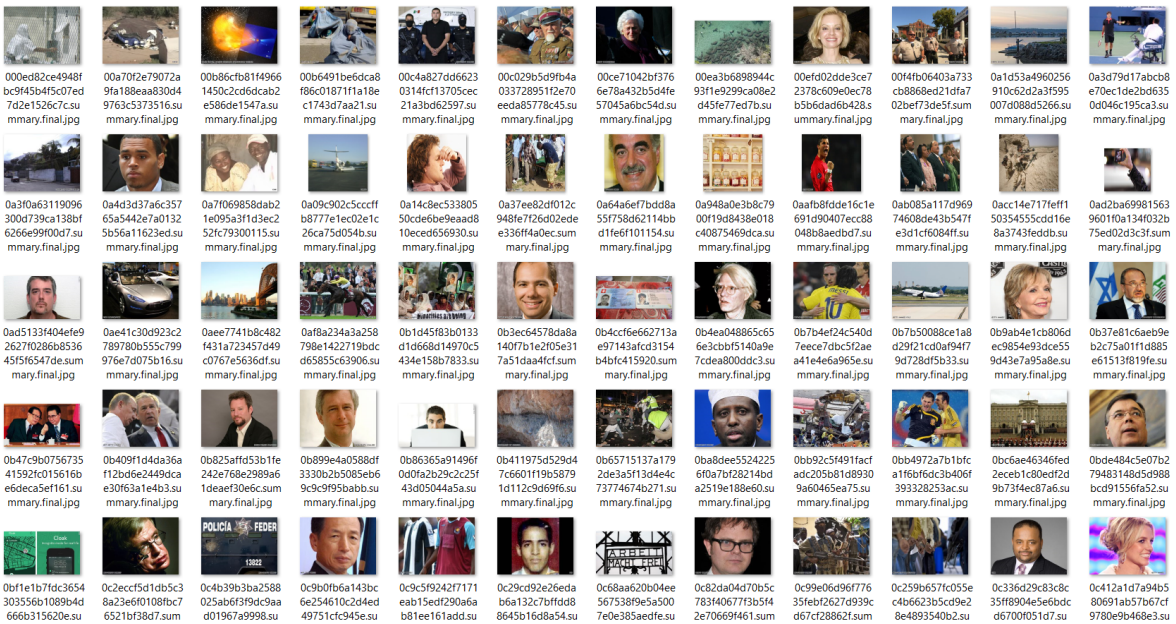

Figure 4.3. A view from the gathered image dataset

a sentence of the news text is similar to the image caption, this sentence is also relevant to the image. The sentences of news articles are labelled as 1 (similar - related) or 0 (not similar – not related) according to this relevance. An example of a news article with its image and image caption is seen on Fig. 4.4. The sentences that are numbered as 1 and 3 are similar to the image caption. Therefore, these are labelled as 1 and the other sentences are labelled as 0. In the dataset, we will have two classes labeled as 0 and 1.

1- **(CNN)** -- British model and television personality Katie Price, also known as Jordan, and her singer husband, Peter Andre, are to separate, according to a statement released Monday.

Peter Andre and Katie Price, who ran the London Marathon last month, are separating.

2- The couple found romance on the reality show "I'm A Celebrity... Get me Out of Here!," which was filmed in the Australian jungle.

3- The statement said: "Peter Andre and Katie Price are separating after four-and-a-half years of marriage," the British Press Association reported.

4- "They have both requested that the media respect their families' privacy at this difficult time."

5- Only last month the couple, whose reality TV show of their life features on British television, said they were trying for another child.

6- They have two children together, son Junior, 3, and 1-year-old daughter Princess Tiáamii.

7- Price has a 6-year-old son Harvey, by footballer Dwight Yorke, who is disabled.

8- Price first made her name as a tabloid newspaper topless model, but has since gone on to become a television star, author and clothes designer.

9- She also competes in show jumping events and has her own stable of horses.

10- Andre, who was born in London but raised in Australia, came to prominence in 1996 with his international hit "Mysterious Girl."

Figure 4.4. Example of a news article with image and image caption

We use three different approaches, the details of which are described in Section 3.1, to automatically understand the similarities of a sentence and caption. We chose the threshold values of similarities for every method by reading the sentences of the news articles ourselves. We labeled the sentences of 50 news articles as being similar to the image caption or not and validated the threshold values by comparing the accuracy with our labeled sentences. If the similarity measure of two sentences is greater than

the threshold, these sentences are similar. In Narayan *et al.* [56], the dataset has also the tokenized versions of the sentences and captions. We used the tokenized version to detect the words more clearly.

The first method is a simple version of Tf-Idf (Term Frequency–Inverse Document Frequency). We count the occurrences of every word in both sentences. The counts of every word construct a vector and we calculate the cosine similarities of them. An example of similarities of a news article is shown in Figure 4.5. The threshold value is determined as 0.35. Threshold-accuracy graph is shown in Figure 4.6 and threshold Receiver Operating Characteristics (ROC) curve is shown in Figure 4.7. Tf-idf measurements are calculated by using Java and Matlab languages in this work.

The second method is using the sentence-to-sentence similarity calculation application of SEMILAR (SEMantic simILARity toolkit). We used the sentence-to-sentence similarity method which calculates the optimum matching by using the Wordnet LESK-Tanim method for word-to-word similarity. An example of similarities of a news article is shown in Figure 4.8. The threshold is 0.25 for this method. Threshold-accuracy graph is shown in Figure 4.9 and threshold Receiver Operating Characteristics (ROC) curve is shown in Figure 4.10. SEMILAR sentence-to-sentence Wordnet Lesk Tanim method is implemented in Java language by using SEMILAR API's Semilar-1.0.jar.

The third method is using the Word2Vec vectors of sentences. As we originated our dataset from the dataset of [56] for text, we used the same technique for text representation and used Word2Vec. This method represents every word as a vector called "word embeddings" that symbolizes semantic relationships. As given in [56], we used the same vector space of word embeddings data created by using the Word2vec skip-gram model with context window size 6, negative sampling size 10, hierarchical softmax and word embeddings size is 200. For representing a sentence, we used the method of averaging the Word2Vec vectors of the words which construct that sentence. The cosine similarity of image caption sentence vector and one of the news text vector is calculated. We explored that this cosine similarities do not show the similarity of two sentences well. Accordingly, we removed the stop words like "a", "and", "but", "how",

Image caption: president bush chats with russian prime minister vladmir putin at the start of the olympic opening ceremonies .	
Similarity	**Sentence**
0.14719415945162714	washington – president bush condemned the escalated violence between russia and u.s. - backed georgia on sunday , while vice president dick cheney said aggression against georgia " must not go unanswered
0.9816513761467889	" president bush chats with russian prime minister vladmir putin at the start of the olympic opening ceremonies
0.14187569991732196	" my administration has been engaged with both sides of this trying to get a ceasefire , " bush told nbc 's bob costas in an interview in beijing , china , where the president has attended olympic events
0.3529139195894344	bush was filmed speaking to russian prime minister vladimir putin during friday 's opening ceremonies and said sunday that he " was firm with vladimir putin " and that " this violence is unacceptable
0.03651678478080381	" violence has continued to rage between russia and the western ally since thursday , when georgia launched an operation to crack down on separatists in south ossetia territory
0.037389709664773764	russia said it wanted to protect its peacekeepers already in south ossetia following ceasefires in years past
0.0	but georgia called it a full - on invasion
0.11643219199555223	and while russia has accused georgia of a genocidal plot to cleanse the region of ethnic ossetians loyal to russia , georgia accuses russia of executing a long - planned war with the aim of taking control of the region – including a key pipeline that carries asian oil to black sea ports
0.11009174311926605	" i expressed my grave concern about the disproportionate response of russia , " bush said of his talk with putin
0.0	" we strongly condemn bombing outside of south ossetia
0.0908673799223074	" putin says he 's concerned about the flood of refugees arriving in russia from south ossetia
0.07311758865129639	russian officials said more than 30,000 refugees have left south ossetia and crossed into russia over the past two days , interfax reported
0.1306590472424788	" the actions of the georgian authorities in south ossetia are a crime , of course , primarily a crime against their own people , " putin said , according to russian news agency interfax
0.08277933418071734	meanwhile , cheney talked to georgian president mikheil saakashvili on sunday , telling him that russia 's aggression against georgia " must not go unanswered
0.1043766995686491	" cheney 's spokeswoman lea ann mcbride said the vice president spoke to saakashvili to express " the united states ' solidarity with the georgian people and their democratically elected government in the face of this threat to georgia 's sovereignty and territorial integrity
0.05388193823456181	" georgia withdrew its forces sunday and offered a ceasefire , which russia refused
0.14598680520314275	" the vice president told president saakashvili that russian aggression must not go unanswered , and that its continuation would have serious consequences for its relations with the united states , as well as the broader international community , " mcbride said
0.0494619366829498	saakashvili has called on the united states and the world community to stop the " intervention and invasion of my sovereign country
0.06425294053171789	" " i think the u.s. is the most powerful country in the world , " he told cnn
0.03238043840981863	" i think the u.s. has lots of leverage
0.0	and i think there are lots of diplomatic means that it could be done through
0.0540628304876326	" two senior officials have told cnn the united states sent envoy matt bryza to the region to help with mediation .

Figure 4.5. Example of Tf-idf method threshold

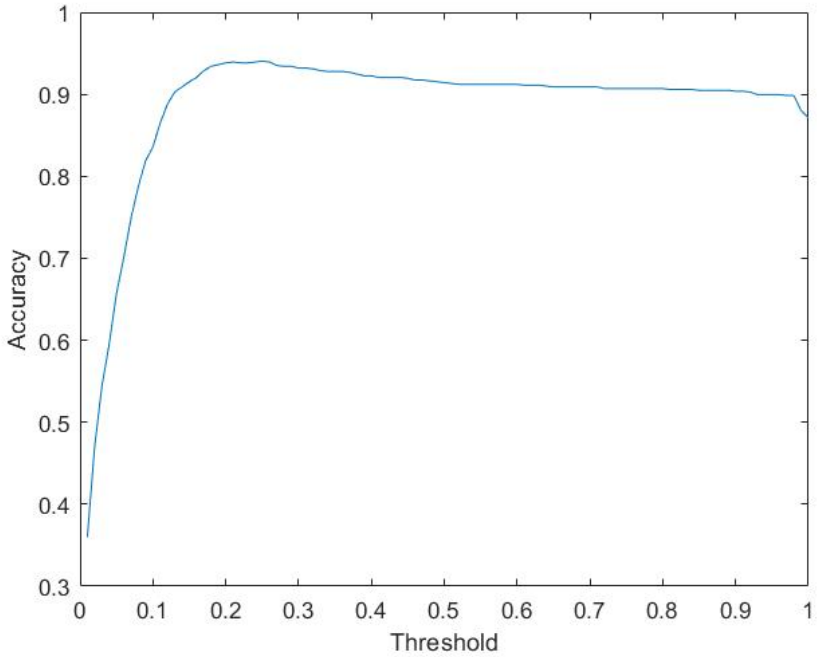

Figure 4.6. Tf-idf method threshold accuracy graph

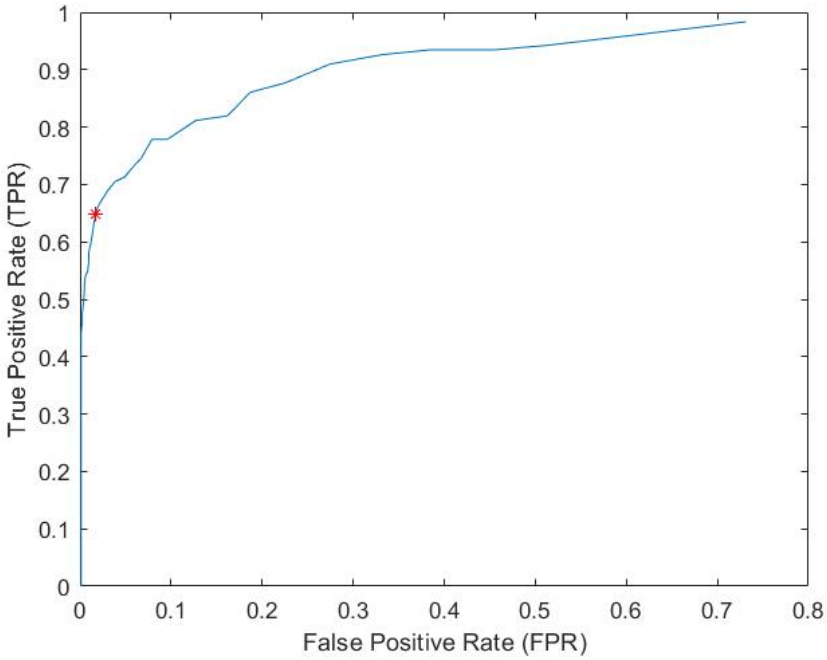

Figure 4.7. Tf-idf method ROC curve

Image caption: peter andre and katie price , who ran the london marathon last month , are separating .	
Similarity	**Sentence**
0.4	– british model and television personality katie price , also known as jordan , and her singer husband , peter andre , are to separate , according to a statement released monday
0.13282542	the couple found romance on the reality show " i 'm a celebrity ... get me out of here ! , " which was filmed in the australian jungle
0.47619048	the statement said : " peter andre and katie price are separating after four - and - a - half years of marriage , " the british press association reported
0.117092945	" they have both requested that the media respect their families ' privacy at this difficult time
0.10526316	" only last month the couple , whose reality tv show of their life features on british television , said they were trying for another child
0.0	they have two children together , son junior , 3 , and 1 - year - old daughter princess tiÃ¡mii
0.11764706	price has a 6 - year - old son harvey , by footballer dwight yorke , who is disabled
0.0952381	price first made her name as a tabloid newspaper topless model , but has since gone on to become a television star , author and clothes designer
0.0	she also competes in show jumping events and has her own stable of horses
0.2	andre , who was born in london but raised in australia , came to prominence in 1996 with his international hit " mysterious girl

Figure 4.8. Example of similarities of SEMILAR method and threshold

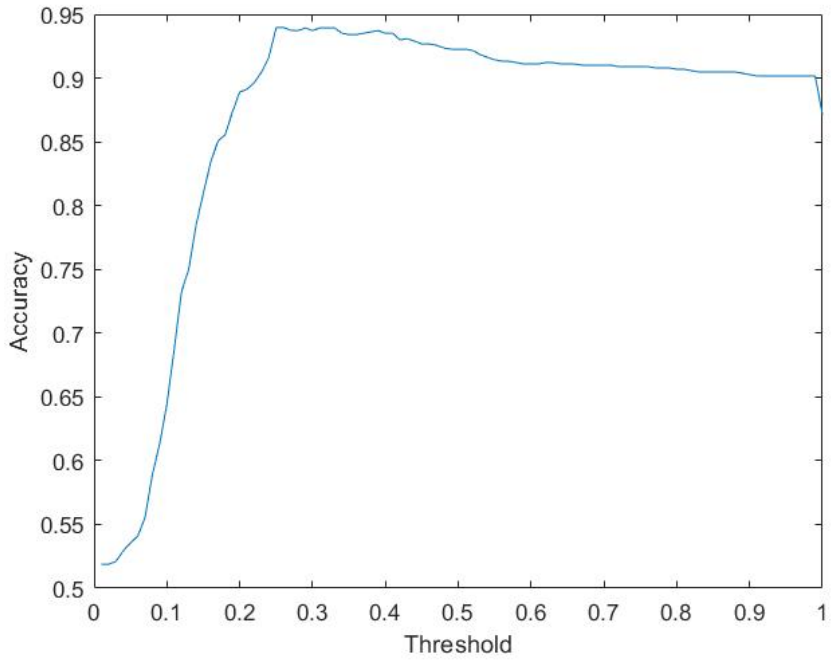

Figure 4.9. SEMILAR method threshold accuracy graph

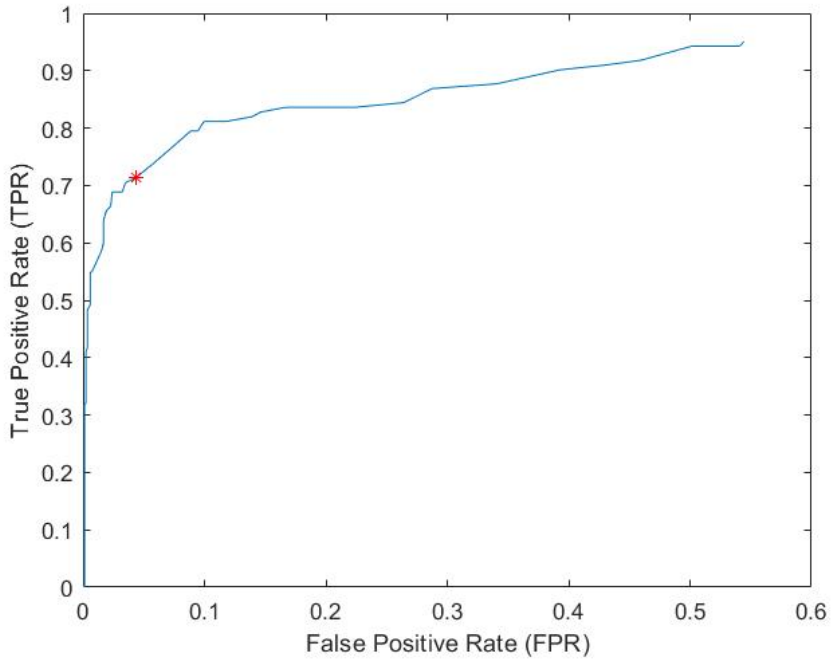

Figure 4.10. SEMILAR method ROC curve

"or". After stop words removal, we created the Word2Vec vectors and calculated cosine similarities of sentences again. The threshold value is set to 0.82. Threshold-accuracy graph is shown in Figure 4.13 and threshold Receiver Operating Characteristics (ROC) curve is shown in Figure 4.14. Two examples of Word2Vec similarity comparing method is shown in Figure 4.11 and Figure 4.12. As seen in Figure 4.11, the similarities of sentences are close to each other and it is difficult where to cut the threshold. But as seen in Figure 4.12, when the stop words are removed from the sentences, the similarities of sentences are more distinguishing. We used Matlab2017b version and matlab functions to process Word2Vec implemententation.

The above mentioned techniques are done for determining the classes for the training and test set. After automatic labeling, we controlled the labels of the sentences by reading and observing the images ourselves in addition.

In conclusion, all the dataset is labelled as 1 (related - similar) and 0 (not related - not similar). 1 and 0 are two classes of this work. The next step is trying to classify

Image caption: a detainee is seen through a fence in july at the u.s. prison camp at guantanamo bay , cuba .	
Similarity	**Sentence**
0.885268	washington – defense secretary robert gates has asked pentagon staff to draw up plans for shutting the u.s. prison camp at guantanamo bay , cuba , a pentagon spokesman said
0.822470	the camp holds about 250 suspected terrorists , down from a peak of roughly 750 men from 40 countries
0.841033	it houses several top al qaeda figures , including khalid sheikh mohammed – the confessed architect of the september 11 , 2001 , attacks
0.857046	gates " has asked his team for a proposal on how to shut it down , what would be required specifically to close it and move the detainees from that facility , while at the same time , of course , ensuring that we protect the american people [from] some very dangerous characters , " pentagon press secretary geoff morrell said thursday
0.830519	morrell described it as a contingency plan in case the new administration wants to take it up early in the new year
0.846101	president - elect barack obama has pledged to close the camp at guantanamo but has n't set a specific timetable
0.730273	gates will continue as defense secretary when obama takes office
0.783048	" i would like to see it closed , " gates told charlie rose in a pbs interview
0.748269	" and i think it will be a high priority for the new administration
0.860164	" officials close to the obama team said in november that the incoming administration is pondering options , including trying some of the guantanamo bay inmates in federal courts , setting up a special national security
0.837025	in an october 31 interview with cnn , obama said only that he would close the facility " as quickly as we can do prudently
0.714021	" " i am not going to give a time certain because i think what we have to do is evaluate all those who are still being held in gitmo , " he said
0.793187	" we have to put in place appropriate plans to make sure they are tried , convicted and punished to the full extent of the law , and that 's going to require , i think , a review of the existing cases , which i have not had the opportunity to do
0.843558	" in may , gates told a senate committee that efforts to shut down the facility were " stuck " over what to do with the inmates .

Figure 4.11. Similarities of Word2Vec method example

Image caption: a detainee is seen through a fence in july at the u.s. prison camp at guantanamo bay , cuba .	
Similarity	**Sentence**
0.837160	washington – defense secretary robert gates has asked pentagon staff to draw up plans for shutting the u.s. prison camp at guantanamo bay , cuba , a pentagon spokesman said
0.732363	the camp holds about 250 suspected terrorists , down from a peak of roughly 750 men from 40 countries
0.782395	it houses several top al qaeda figures , including khalid sheikh mohammed – the confessed architect of the september 11 , 2001 , attacks
0.785269	gates ” has asked his team for a proposal on how to shut it down , what would be required specifically to close it and move the detainees from that facility , while at the same time , of course , ensuring that we protect the american people [from] some very dangerous characters , ” pentagon press secretary geoff morrell said thursday
0.694846	morrell described it as a contingency plan in case the new administration wants to take it up early in the new year
0.736200	president - elect barack obama has pledged to close the camp at guantanamo but has n’t set a specific timetable
0.614514	gates will continue as defense secretary when obama takes office
0.692013	” i would like to see it closed , ” gates told charlie rose in a pbs interview
0.559865	” and i think it will be a high priority for the new administration
0.790099	” officials close to the obama team said in november that the incoming administration is pondering options , including trying some of the guantanamo bay inmates in federal courts , setting up a special national security court to deal with cases involving the most sensitive intelligence information , and releasing some inmates
0.798074	in an october 31 interview with cnn , obama said only that he would close the facility ” as quickly as we can do prudently
0.600605	” ” i am not going to give a time certain because i think what we have to do is evaluate all those who are still being held in gitmo , ” he said
0.680755	” we have to put in place appropriate plans to make sure they are tried , convicted and punished to the full extent of the law , and that ’s going to require , i think , a review of the existing cases , which i have not had the opportunity to do
0.735806	” in may , gates told a senate committee that efforts to shut down the facility were ” stuck ” over what to do with the inmates .

Figure 4.12. Similarities of Word2Vec method without stop words

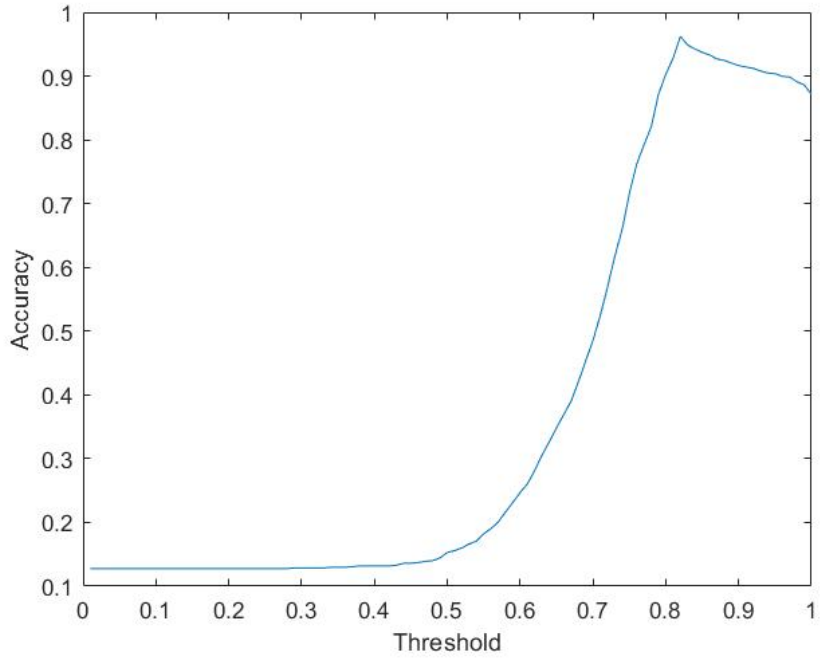

Figure 4.13. Word2Vec method threshold accuracy graph

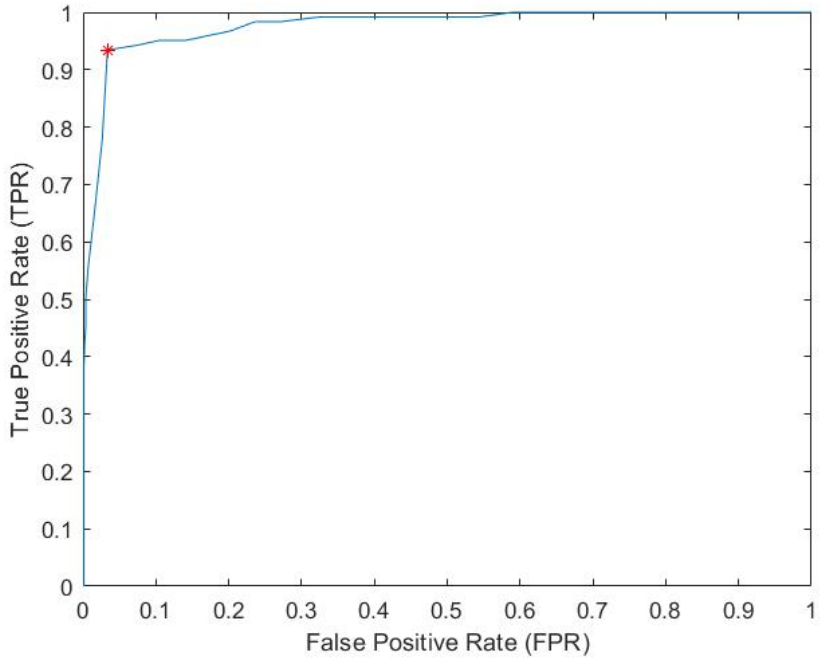

Figure 4.14. Word2Vec method ROC curve

these classes.

5. EXPERIMENTS AND RESULTS

In this chapter, feature extraction techniques of image and text, classification methods and classification test results with these feature vectors prepared on three different dataset are described. As declared in Chapter 4, we have two classes and we try to classify these two classes to understand if a news image is related to a news article text sentence. For this purpose we need to represent image and text parts of the news articles and apply classification techniques.

5.1. Image Feature Extraction

Our purpose in this work is to identify the news image related sentences of the news article. Hence, we need to represent the images as vectors to make them meaningful for classification methods. For this purpose, we extract the features of images to combine the relationship of images with text sentences. Image feature extraction is implemented by using Histogram of Oriented Gradients (HOG) and Bag Of Visual Words (BOVW) methodologies that have the details mentioned in Section 3.2. In this chapter, we give the implementation details of these feature extraction methods. We use Matlab2017b library to extract the HOG and BOVW features of images.

5.1.1. Histogram of Oriented Gradients (HOG)

In this study, we implement the HOG descriptor representation by dividing the image into cells. Sizes of cells are 32*32. Each cell is discretized into 9 angular bins. The block size of HOG feature vector is 2*2. Histogram of gradient directions for the pixels within the cell is computed for each cell. The image sizes are mostly 219x292 in the dataset. If an image which is in the small portion has different sizes, we resized it to 219x292 sizes to get the same number of features like the most of the images. After implementing the HOG descriptor algorithm, final feature vector length for each image in the dataset is 1440.

5.1.2. Bag Of Visual Words (BOVW)

In this study, we implement the BOVW descriptor by using Speeded-Up Robust Features (SURF) feature extraction technique to detect the key points. After SURF, we cluster these SURF features to construct the vocabulary. As mentioned in Section 3.2.2, for k-means clustering, k represents the vocabulary size. To compare and find the better implementation results, we selected three different k values. At the last step, the counts of the words in an image forms the histogram vector of words that represents the image. We used 500, 1000 and 2000 sized vocabulary and histogram vector on this work for comparison.

5.1.3. Principal Component Analysis (PCA)

In this work, HOG and BOVW image feature description techniques are used and the vector lengths of these methods are 1440 and 500, 1000, 2000 respectively. On the other hand, the text features are represented by Word2Vec vectors that have 200 features. This situation shows that image features are much more than the text features, so image features may have more weight than the text features. On account of this, we decided to test these features with equal weights also. To fulfill this decision, we used the PCA technique just for image features. PCA is implemented by using WEKA tool.

HOG features vectors have 1440 length, we applied the parameter variance covered as 0.89 to have 200 features. For 500, 1000 and 2000 lengths BOVW vectors, the variance value is 0.64, 0.48 and 0.31.

5.2. Text Feature Extraction

Our purpose being to identify the news image related sentences of news article, we need to represent the sentences of the news article as vectors to make them meaningful for classification methods, the same as the image representation. As mentioned in Chapter 4 part of this work, we used Word2Vec word embedding vectors to implement

the sentences. Because we used the dataset of Narayan *et al.* [56], we used the same vector space of word embeddings data created by using the Word2vec skip-gram model with context window size 6, negative sampling size 10 and hierarchical softmax. Word embeddings size in our study is 200. Actually, Word2Vec vectors represents the words of the dataset. For each different word, we have a different vector. In this work, we need to represent the sentences to understand their relationship with the images. Therefore, we used the technique of averaging the words of a sentence. Getting the average vector of words contained in the sentence symbolizes the vector of the sentence. As referred in Chapter 4, we observed that getting all the words average does not represent the sentence well. By removing the commonly used words, the stop words such as "a", "and", the more meaningful remaining words represent the sentences better as shown in Chapter 4 in Figure 4.12. After this observation, we used stop words removed Word2Vec vectors. We use Matlab2017b library to implement Word2Vec process.

5.3. Creating Training and Test Sets

In this work, the aim is to select the news article sentences which are related to the news image. In order to succeed in this aim, we extracted the image and text features as seen on previous sections. After obtaining these features, we united image and text features to cultivate them together. Both image features and text features of a sentence are concatenated to create a new feature vector. For example, HOG features are 1440 sized vectors and Word2Vec features are 200 sized vectors. When these two vectors are combined together, we have a new 1640 sized vector, and 700, 1200 and 2200 sized vectors for BOVW. This single vector represents the unity of the image and that individual sentence. For all sentences of a news article, the image features and that sentence features are concatenated repeatedly. As seen in Chapter 4, the image related sentences are labelled as 1 and image unrelated sentences are labelled as 0. This is represented by the concatenated single vector of image and text features together and that single vector is labelled as 1 or 0. By these operations, the last numerical dataset that is needed for classification is acquired.

As mentioned in Chapter 4, Narayan *et al.* [56] already splitted the dataset as train and test. We firstly used these training and test sets. Afterwards, we reconstructed these datasets and reconstruction details are mentioned in the Experiments and Results Section 5.4.

For the classification process, we use WEKA tool and this tool uses a special file format called "arff" for training and test sets. This arrf formatted files that include the single vectors of image features and text features are prepared by using Java language.

5.4. Experiments and Results

In this part we mention about the classification techniques used to select the image related sentences in our three datasets and result of our experiments.

5.4.1. Classification Methods

In this study, we try to find the image related sentences of the news articles. As mentioned in Chapter 4, the related and not related sentences are labelled as 0 and 1, so we train our methods for two classes. The methods of classification we use are Naive Bayes, k-Nearest Neighbors (k-NN) and Random Forest Decision Tree classifiers. The details of the classification methods are explained in Section 3.4. As mentioned in Section 5.3, the dataset is divided as train and test set. The classifiers are modelled with the training set and experimented on the test set. For training, the train dataset is divided into stratified 5-fold cross-validation and tested with this version also. These classification methods have been developed on Java programming language using the WEKA library.

Naive Bayes classification works on probabilities of the features with each other, so we directly used the WEKA library without any addition on the method.

The IBk algorithm used in the WEKA library was used to classify with k-NN. The IBk algorithm uses the k parameter when classifying. The k parameter can take

between 1 and 64 values. For each k value in this value range, the best k value is determined by taking the classification results. When k = 1, the best classification results were obtained.

Random Forest algorithm on WEKA has tree depth and number of trees parameters that can change the results. According to our experiments, when the tree number is 100 and tree depth is 35, the best results are taken.

5.4.2. Results

Experimental results are taken for 5-fold cross-validation and test datasets. The detailed results are shown in Chapter A and the summarized results are shown in this section. In Table A.1 through Table A.6, the performance of classifiers is reported according to the training, image and text representation techniques. Tables present the comparison of Naive Bayes, k-NN and Random Forest classification methods, image representation methods of HOG, BOVW-500, BOVW-1000 and BOVW-2000 and in terms of Precision, Recall, F1 measure, PRC-area and ROC-area on three datasets that are prepared by using Tf-idf, SEMILAR and Word2Vec similarity comparison methods. Each dataset that are created by Tf-idf and Word2Vec cosine similarities, SEMILAR application sentence similarity which are used for labeling image-related and irrelevant sentences in the dataset are seperately classed with each image identifier. Table A.1 and Table A.2 present the results for the training set that is labelled by tf-idf cosine similarity method. Table A.3 and Table A.4 present the results for the training set that is labelled by SEMILAR sentence-to-sentence similarity method. Table A.5 and Table A.6 present the results for the training set that is labelled by Word2Vec cosine similarity method. Furthermore, since image descriptors have longer vectors than text descriptors, they are tested by balancing the weights of the text and image descriptor vectors by reducing the vector size by Principal Component Analysis (PCA) method and the results are shown through Table A.7 and Table A.12. Table 5.1 and Table 5.2 presents the summarization of the other detailed tables in Recall results with cross validation results and test dataset validation results. These summary tables just show the recall results for class 1 because of searching for image related sentences.

Table 5.1. Summarization table for 5-fold cross validation results

Dataset / Image Representor	**Classification Method(No PCA/PCA)**					
	k-NN		**Naive Bayes**		**Random Forest**	
Tf-Idf	*0,773*	*0,750*	*0,692*	*0,704*	*0,411*	*0,399*
HOG	0,750	0,750	0,672	0,704	0,411	0,396
BOVW-500	0,750	0,750	0,692	0,703	0,386	0,398
BOVW-1000	0,750	0,750	0,664	0,703	0,377	0,399
BOVW-2000	0,750	0,750	0,626	0,704	0,371	0,399
SEMILAR	*0,724*	*0,724*	*0,666*	*0,674*	*0,397*	*0,402*
HOG	0,717	0,717	0,644	0,674	0,397	0,395
BOVW-500	0,724	0,724	0,666	0,674	0,390	0,400
BOVW-1000	0,724	0,724	0,642	0,673	0,384	0,401
BOVW-2000	0,724	0,724	0,605	0,674	0,384	0,402
Word2Vec	*0,798*	*0,868*	*0,732*	*0,747*	*0,396*	*0,409*
HOG	0,798	0,868	0,732	0,747	0,303	0,409
BOVW-500	0,867	0,867	0,729	0,746	0,396	0,407
BOVW-1000	0,867	0,867	0,704	0,746	0,392	0,407
BOVW-2000	0,867	0,867	0,653	0,746	0,392	0,406

Table 5.2. Summarization table for test validation results

Dataset / Image Representor	**Classification Method(No PCA/PCA)**					
	k-NN		**Naive Bayes**		**Random Forest**	
Tf-Idf	*0,411*	*0,376*	*0,746*	*0,782*	*0*	*0,056*
HOG	0,108	0,152	0,450	0,782	0	0,056
BOVW-500	0,216	0,350	0,667	0,777	0	0,056
BOVW-1000	0,411	0,360	0,746	0,777	0,020	0,046
BOVW-2000	0,369	0,376	0,604	0,772	0	0,051
SEMILAR	*0,354*	*0,365*	*0,760*	*0,804*	*0,026*	*0,070*
HOG	0,261	0,299	0,442	0,804	0	0,055
BOVW-500	0,216	0,288	0,704	0,804	0,020	0,070
BOVW-1000	0,354	0,365	0,760	0,804	0,026	0,066
BOVW-2000	0,276	0,295	0,633	0,804	0	0,063
Word2Vec	*0,395*	*0,383*	*0,868*	*0,891*	*0,051*	*0,078*
HOG	0,395	0,383	0,811	0,891	0,005	0,056
BOVW-500	0,371	0,307	0,868	0,868	0,036	0,062
BOVW-1000	0,354	0,370	0,786	0,868	0,051	0,078
BOVW-2000	0,279	0,370	0,756	0,875	0,005	0,070

The first results show us that the HOG method with Naive Bayes classifier delivers the overall best performance with precision, recall and f-measure values on the test dataset among the combinations of image features and Word2Vec text feature when the training dataset ground truth is labelled by using the Word2Vec cosine similarities of the sentences. The training set that is labelled with Word2Vec cosine similarities is giving better performance than SEMILAR and tf-idf in all classifier methods. Because of considering just the number of words, not semantic meaning of words, tf-idf is the worst method for ground-truth labelling of the sentences. BOVW methods also give good performance rate with k-Nearest Neighbors method. Random Forest classifier method gives poor results on test set. The precision values of Random Forest classifier are good, but recall is so low or 0 on test. Random Forest can not find any results for class 1. These are our first comments about the results at first glance.

As seen on the Table 5.2 recall values, although the results are good for Naive Bayes classifier, on other classifiers the results are so low. Through Table A.1 - Table A.12, although k-NN method has good precision and recall values on 5-folds tests, it is not good at test set. This may be due to the shortage of test dataset, 5-folds validation has more examples for testing. If we evaluate these results, we think that these low results are because of having an unbalanced dataset. The number of image related sentences on a news article is too limited when compared with the unrelated sentences. The related sentences are approximately one-tenth among all sentences. Because the 1 and 0 classes are not balanced, the classifiers tend to classify the 1 labelled sentences as 0. Based on this inference, we tried to augment the samples of class 1 or to weight the classes to take better results.

First, to augment class 1, we used WEKA's Synthetic Minority Oversampling TEchnique (SMOTE) [58] method for resampling. SMOTE is an oversampling procedure. It creates synthetic samples by selecting several similar instances using the distance metric of k nearest neighbors and perturbing an instance one attribute at a time by a random amount within the difference to the neighboring instances. SMOTE generates synthetic samples from the minor class instead of creating copies. We used SMOTE technique to oversample the instances of class 1 for training. At last, the

number of instances of class 0 and class 1 were equal to each other.

Second technique we used for class unbalancing is WEKA's ClassBalancer. ClassBalancer reweights the instances in the dataset to make each class at the same total weight.

SMOTE oversampling and class unbalancing techniques increased the results of 5-fold cross validation but did not make any change for the experiments on the test set. Because test set has not changed and and it still has the same number of instances for both classes. When we classify the instances, they are spreaded to the same classes like previous experiments. Therefore, we did not utilize these two techniques.

Another method we used for unbalancing problem is undersampling. Undersampling takes a random subsample of the dataset. We selected equal number of instances for both classes. This technique did not increase our results both on 5-fold cross validation and test datasets. Because on undersampling, we do not have much more examples for class 0 and this factor decreases the cross validation and test results.

Lastly, we tried another methodology. The training dataset has 152.356 samples but test dataset has 1.325 samples. In our experiments, we realized that test dataset is too small when compared to the training dataset and results on test set does not give good results. While augmenting the main dataset of Narayan *et al.* [56], we could not use some of the news as mentioned in Chapter 4. So, their test set became smaller on our version. With this point of view, we decided to experience new results by dividing the dataset as train and test again. We first merged training and test sets and splitted the sets again. Test set is one out of five of the all instances. The new results with the new training and test sets is through Table A.13 an Table A.18. We also test with PCA applied versions again and the results are through Table A.19 and Table A.24. Table 5.3 and Table 5.4 present the summarization of the other detailed tables in Recall results with cross validation results and test dataset validation results. These summarization tables point the recall measures for class 1 because of searching for image related sentences.

Table 5.3. Summarization table for 5-fold cross validation results with new split

Dataset / Image Representor	**Classification Method(No PCA/PCA)**					
	k-NN		**Naive Bayes**		**Random Forest**	
Tf-Idf	*0,725*	*0,725*	*0,692*	*0,706*	*0,493*	*0,348*
HOG	0,725	0,725	0,673	0,706	0,358	0,345
BOVW-500	0,725	0,725	0,692	0,705	0,493	0,345
BOVW-1000	0,725	0,725	0,662	0,706	0,322	0,346
BOVW-2000	0,725	0,725	0,620	0,705	0,319	0,348
SEMILAR	*0,725*	*0,702*	*0,665*	*0,677*	*0,353*	*0,350*
HOG	0,700	0,700	0,647	0,675	0,353	0,349
BOVW-500	0,702	0,702	0,665	0,676	0,335	0,348
BOVW-1000	0,725	0,702	0,645	0,676	0,329	0,35
BOVW-2000	0,725	0,702	0,603	0,677	0,325	0,345
Word2Vec	*0,857*	*0,857*	*0,735*	*0,747*	*0,381*	*0,364*
HOG	0,857	0,857	0,735	0,747	0,381	0,364
BOVW-500	0,857	0,857	0,732	0,746	0,343	0,357
BOVW-1000	0,857	0,857	0,699	0,746	0,339	0,359
BOVW-2000	0,857	0,857	0,650	0,747	0,335	0,358

Table 5.4. Summarization table for test validation results with new split

Dataset / Image Representor	**Classification Method(No PCA/PCA)**					
	k-NN		**Naive Bayes**		**Random Forest**	
Tf-Idf	*0,550*	*0,543*	*0,676*	*0,693*	*0,336*	*0,349*
HOG	0,469	0,489	0,671	0,690	0,319	0,346
BOVW-500	0,505	0,536	0,676	0,693	0,336	0,345
BOVW-1000	0,550	0,543	0,664	0,691	0,330	0,349
BOVW-2000	0,476	0,525	0,623	0,692	0,323	0,348
SEMILAR	*0,521*	*0,538*	*0,667*	*0,670*	*0,342*	*0,355*
HOG	0,521	0,538	0,646	0,670	0,315	0,350
BOVW-500	0,494	0,515	0,667	0,669	0,342	0,355
BOVW-1000	0,515	0,523	0,656	0,668	0,334	0,349
BOVW-2000	0,468	0,490	0,619	0,667	0,327	0,354
Word2Vec	*0,552*	*0,559*	***0,732***	***0,742***	*0,343*	*0,357*
HOG	0,545	0,559	**0,732**	**0,742**	0,307	0,355
BOVW-500	0,552	0,548	0,725	0,74	0,343	0,357
BOVW-1000	0,518	0,551	0,69	0,741	0,333	0,354
BOVW-2000	0,492	0,542	0,632	0,737	0,321	0,357

On the latest results, it is observed in Table 5.4 that Naive Bayes classifier, which uses HOG image and Word2Vec text identifiers on the dataset, whose sentence and image relevance is tagged by calculating the cosine similarity between Word2Vec sentences, gives the best result. The dataset which is labelled by using Word2Vec cosine similarity gives better results on all classifier methods than SEMILAR and Tf-idf. Because of considering just the number of words, not semantic meaning of words, tf-idf is the worst method for ground-truth labelling of the sentences. Although both Word2Vec and SEMILAR methods try to find a semantic similarity, Word2Vec has given better similarity results because Word2Vec vectors are created specifically for the news dataset and SEMILAR is calculating similarity independent to the news dataset. The HOG method best describes the images, while the Word2Vec method best reflects the meaning of sentences. While the Naive Bayes classifier gives the best results in this problem solving, it can be said that the Random Forest classifier is not a suitable classifier for this problem since it gives the worst results. In PCA applied versions, the results are always better. Identifying important features with PCA, improves problem solving and balances image and text properties. Naive Bayes is a method that is widely used in computer vision and it is a probabilistic method working with the assumption that features and classes are uncorrelated of each other. The image features and properties are uncorrelated so Naive Bayes have better results. The PCA method has also led to better results on Naive Bayes classifier because PCA transforms the input data into a new space with new uncorrelated dimensions. The Random Forest classifier is a more useful method for multiple classes instead of binary. In a news article, sentences that are not related to the image are 10 times more than the related sentences. Since Random Forest classifier divides train data into subdivisions and creates many trees from them, a small number of class 1 of data cannot be sufficiently placed on the leaves of trees. As there is no balanced distribution between classes 0 and 1, classifier methods such as Random Forest try to classify the samples belonging to class 1 as class 0.

HOG gives better results on Naive Bayes while BOVW gives better results with k-NN on Table 5.3 recall values and on Table A.18 and Table A.24 precision, recall and f-measure values. In later studies, these two methods can be combined to achieve

better results. Experiments can also be carried out with different color and texture image identifiers.

6. CONCLUSION

Identifying image related sentences and image caption generation for news articles has become a remarkable research area with the enormous amount of data and is a very interesting problem for combining aspects of computer vision and natural language processing. In this study, we try to identify the image related sentences by using the news article text content.

In this study, a model which tries to find the sentence set from the sentences in news articles related to the news image is suggested. The CNN news dataset which contains just the text of news article and captions of images is evolved by collecting the news images from web to perform our novel proposed model. Image and text descriptors have been extracted to establish the connection between news sentences and images. Extracted image features are Histogram of Oriented Gradients (HOG), Bag Of Visual Words (BOVW) with 500, 1000 and 2000 words. Text features are extracted by using the mean of Word2Vec 200 sized word vectors. These features are classified as 1 and 0 according to their relatedness for ground truth by using three different ways that are cosine similarity of Tf-Idf, sentence-to-sentence similarity measurement of SEMILAR and cosine similarity of Word2Vec vectors. The arranged dataset helps us to track the relationship between news image features and text features of sentences of news articles. Naive Bayes, k-Nearest Neighbors and Random Forest classifiers are used as classification methods. HOG has 1440 and BOVW has 500, 1000 and 2000 features. When compared to 200 sized text features, image features may have more weight on the results. To reduce this probability, we applied Principal Component Analysis (PCA) to the image features to make them 200-sized like text features. These results are compared in the results section of this study. The first results help us to discover that 1 class has one-tenth instance number than 0 class. Test dataset has also very little instances and this status exposes poor results on classification. To solve unbalancing problem of the binary classed dataset, we tried different procedures but did not help because of not changing test dataset unbalance. At the same time, because the test dataset is too small we have poor results, we decided to recreate the test and train

datasets. We assembled training and test sets and divided them again to enhance the test set. We implemented the experiments again for these new sets.

For identifying the image related sentences in a news article, HOG image features with the combination of Word2Vec text features yields better performance on Naïve Bayes classifier method that is trained with the training dataset which is labelled with the technique of calculating Word2Vec cosine similarities of the sentences.

For future works, we plan to unite the HOG and BOVW image features to achieve better results. Image descriptors such as color, texture, object or deep features different from HOG and BOVW may also support for better results. In later studies, the integration of our new dataset to a deep learning architecture that is used in later works can be examined.

REFERENCES

1. Qimin Cheng, P. F., Qian Zhang *et al.*, "A survey and analysis on automatic image annotation", *Pattern Recognition 79 (2018) 242–259*, 2018.

2. Jia-Yu Pan, P. D., Hyung-Jeong Yang *et al.*, "Automatic Image Captioning", *IEEE International Conference on Multimedia and Expo (ICME)*, 2004.

3. Koen Deschacht, M.-F. M., "Text Analysis for Automatic Image Annotation", *Sixth International Conference on Natural Computation (ICNC 2010)*, 2010.

4. Yu Tang Guo, B. L., "An Automatic Image Annotation Method Based on the Mutual K-Nearest Neighbor Graph", *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2007.

5. Ahmet Aker, R. G., "Generating image descriptions using dependency relational patterns", *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010.

6. Xin Fan, M. T., Ahmet Aker *et al.*, "Automatic Image Captioning From the Web For GPS Photographs", *Proceedings of the 11th ACM SIGMM International Conference on Multimedia Information Retrieval, MIR 2010*, 2010.

7. Laura Plaza, A. A., Elena Lloret, "Improving Automatic Image Captioning Using Text Summarization Techniques", *13th International Conference, Text, Speech and Dialogue*, 2010.

8. May The Yu, M. M. S., "Automatic image captioning system using integration of N-cut and color-based segmentation method", *SICE Annual Conference*, 2011.

9. Alex Krizhevsky, G. E. H., Ilya Sutskever, "ImageNet Classification with Deep Convolutional Neural Networks", *Neural Information Processing Systems(NIPS)*,

2012.

10. Mason, R., "Domain-Independent Captioning of Domain-Specific Images", *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2013.

11. Su Mei Xi, Y. I. C., "Image caption automatic generation method based on weighted feature", *International Conference on Control, Automation and Systems (ICCAS)*, 2013.

12. Mert Kılıçkaya, A. E., Erkut Erdem *et al.*, "Data-driven image captioning with meta-class based retrieval", *Signal Processing and Communications Applications Conference (SIU)*, 2014.

13. Jeff Donahue, M. R., Lisa Anne Hendricks *et al.*, "Long-term Recurrent Convolutional Networks for Visual Recognition and Description", *Computer Vision and Pattern Recognition (CVPR)*, 2015.

14. Hao Fang, F. I., Saurabh Gupta *et al.*, "From Captions to Visual Concepts and Back", *Computer Vision and Pattern Recognition (CVPR)*, 2015.

15. Tsung-Yi Lin, S. B., Michael Maire *et al.*, "Microsoft COCO: Common Objects in Context", *European Conference on Computer Vision*, 2014.

16. Xinlei Chen, T.-Y. L., Hao Fang *et al.*, "Microsoft coco captions: Data collection and evaluation server", *arXiv preprint arXiv:1504.00325*, 2015.

17. Xinlei Chen, C. L. Z., "Mind's Eye: A Recurrent Visual Representation for Image Caption Generation", *Computer Vision and Pattern Recognition (CVPR)*, 2015.

18. C. Rashtchian, M. H., P. Young *et al.*, "Collecting image annotations using Amazon's mechanical turk", *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2010.

19. Andrej Karpathy, L. F.-F., "Deep Visual-Semantic Alignments for Generating Image Descriptions", *Computer Vision and Pattern Recognition (CVPR)*, 2015.

20. Kelvin Xu, R. K., Jimmy Ba *et al.*, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", *International Conference on Machine Learning*, 2015.

21. Quanzeng You, Z. W., Hailin Jin *et al.*, "Image Captioning with Semantic Attention", *Computer Vision and Pattern Recognition (CVPR)*, 2016.

22. Kun Fu, R. C., Junqi Jin *et al.*, "Aligning Where to See and What to Tell: Image Captioning with Region-Based Attention and Scene-Specific Contexts", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

23. Cesc Chunseong Park, G. K., Byeongchang Kim, "Attend to You: Personalized Image Captioning with Context Sequence Memory Networks", *arXiv preprint arXiv:1704.06485*, 2017.

24. Yansong Feng, M. L., "How Many Words is a Picture Worth? Automatic Caption Generation for News Images", *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 1239–1249, Uppsal, Sweden. Association for Computational Linguistics*, 2010.

25. Yansong Feng, M. L., "Automatic Caption Generation for News Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

26. Priyanka Jadhav, R. C., Sayali Joag *et al.*, "Automatic Caption Generation for News Images", *International Journal of Computer Science and Information Technologies (IJCSIT)*, 2014.

27. K.Vijay, D. R., "Generation Of Caption Selection For News Images Using Stemming Algorithm", *International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)*, 2015.

28. Arnau Ramisa, F. M.-N., Fei Yan *et al.*, "BreakingNews: Article Annotation by Image and Text Processing", *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 40, No. 5*, 2018.

29. Laura Hollink, M. v. H., Adriatik Bedjeti *et al.*, "A Corpus of Images and Text in Online News", *International Conference on Language Resources and Evaluation LREC*, 2016.

30. Gerard Salton, H. W., Edward A. Fox, "Extended Boolean information retrieval", *Communications of the ACM*, Vol. 28 (11), p. 1022–1036, 1983.

31. G. Salton, M. J. M., "Introduction to modern information retrieval", , 1986.

32. G. Salton, C. B., "Term-weighting approaches in automatic text retrieval", *Information Processing Management*, Vol. 24 (5), 1988.

33. H. Wu, K. W., R. Luk *et al.*, "Interpreting TF-IDF term weights as making relevance decisions", *ACM Transactions on Information Systems*, Vol. 26 (3), 2008.

34. Luhn, H. P., "A Statistical Approach to Mechanical Encoding and searching of Literary Information", *IBM Journal of Research and Development*, Vol. 1:4, pp. 309–317, 1957.

35. Jones, K. S., "A Statistical Interpretation of Term Specificity and Its Application in Retrieval", *Journal of Documentation*, Vol. 28, pp. 11–21, 1972.

36. *Tf-Idf*, `http://www.tfidf.com/`, accessed at February 2019.

37. Vasile Rus, R. B., Mihai Lintean *et al.*, "SEMILAR: The Semantic Similarity Toolkit", *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, p. 163–168, August 4-9 2013.

38. *SEMILAR: A Semantic Similarity Toolkit*, `http://www.semanticsimilarity.org/`, accessed at February 2019.

39. George A. Miller, C. F., Richard Beckwith, "Introduction to WordNet: An On-line Lexical Database", *Int. J. Lexicograph*, p. 235–244, 1998.

40. *WordNet*, `https://wordnet.princeton.edu/`, accessed at February 2019.

41. Satanjeev Banerjee, T. P., "An adapted Lesk algorithm for word sense disambiguation using WordNet", *Computational Linguistics and Intelligent Text Processing*, Vol. 2276, p. 117–171, 2002.

42. Lesk, M., "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone", *Proceedings of SIGDOC '86*, 1986.

43. Tomas Mikolov, G. C., Kai Chen *et al.*, "Efficient estimation of word representations in vector space", *Proceedings of Int. Conf. Learn. Representations*, 2013.

44. Tomas Mikolov, K. C., Ilya Sutskever *et al.*, "Distributed representations of words and phrases and their compositionality", *Proceedings of the 26th International Conference on Neural Information Processing Systems*, p. 3111–3119, 2013.

45. Quoc Le, T. M., "Distributed representations of sentences and documents", *Proceedings of the 31 st International Conference on Machine Learning*, p. 1188–1196, 2014.

46. *Vector Representations of Words*, `https://www.tensorflow.org/tutorials/representation/word2vec`, accessed at February 2019.

47. *Word2Vec Tutorial - The Skip-Gram Model*, `http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/`, accessed at February 2019.

48. Navneet Dalal, B. T., "Histograms of oriented gradients for human detection", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1.

49. G. Csurka, L. F., C. R. Dance *et al.*, "Visual Categorization with Bags of Keypoints", *Proceedings of ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

50. *Bag of Visual Words in a Nutshell*, `https://towardsdatascience.com/bag-of-visual-words-in-a-nutshell-9ceea97ce0fb`, accessed at February 2019.

51. H. Bay, T. T., A. Ess *et al.*, "SURF:Speeded Up Robust Features", *Computer Vision and Image Understanding (CVIU)*, Vol. 110.

52. *Introduction to SURF (Speeded-Up Robust Features)*, `https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html`, accessed at February 2019.

53. *Feature Detection and Matching*, `https://courses.cs.washington.edu/courses/cse576/13sp/projects/project1/artifacts/woodrc/index.htm`, accessed at February 2019.

54. *k-means clustering*, `https://en.wikipedia.org/wiki/K-means_clustering`, accessed at February 2019.

55. *Naive Bayesian*, `https://www.saedsayad.com/naive_bayesian.htm`, accessed at February 2019.

56. Shashi Narayan, S. B. C., Nikos Papasarantopoulos *et al.*, "Neural Extractive Summarization with Side Information", *Association for the Advancement of Artificial Intelligence*, 2018.

57. K. M. Hermann, E. G., T. Kocisky *et al.*, "Teaching machines to read and comprehend", *Neural Information Processing Systems(NIPS)*, Vol. 28, p. 1693–1701, 2015.

58. Chawla, N. V. *et al.*, "Synthetic Minority Over-sampling Technique", *Journal of*

Artificial Intelligence Research, Vol. 16, pp. 321–357, 2002.

APPENDIX A: TABLES OF EXPERIMENTAL RESULTS

In this chapter, the tables of the detailed experimental results of Naive Bayes, k-NN and Random Forest classifiers with HOG, BOW-500, BOW-1000, BOW-2000 image features for 5-folds cross validation and test dataset are given in Precision, Recall, F-measure, PRC-area and ROC area values on the datasets that are prepared by using Tf-Idf, SEMILAR and Word2Vec similarities.

Table A.1. 5-fold cross validation results on Tf-idf dataset

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,943	0,671	0,784	0,951	0,734
		1	0,201	0,672	0,309	0,291	0,735
		Total	0,862	0,671	0,732	0,879	0,735
	BOVW-500	**0**	0,948	0,695	0,802	0,952	0,753
		1	0,218	0,692	0,331	0,271	0,752
		Total	0,868	0,694	0,75	0,878	0,753
	BOVW-1000	**0**	0,942	0,670	0,783	0,940	0,705
		1	0,198	0,664	0,305	0,200	0,703
		Total	0,861	0,669	0,731	0,859	0,705
	BOVW-2000	**0**	0,930	0,612	0,738	0,924	0,63
		1	0,166	0,626	0,262	0,149	0,627
		Total	0,846	0,614	0,686	0,839	0,629
k-NN	HOG	**0**	0,968	0,934	0,951	0,964	0,842
		1	0,584	0,750	0,657	0,474	0,842
		Total	0,926	0,914	0,919	0,910	0,842
	BOVW-500	**0**	0,968	0,934	0,951	0,964	0,842
		1	0,584	0,750	0,657	0,474	0,842
		Total	0,926	0,914	0,919	0,91	0,842
	BOVW-1000	**0**	0,968	0,934	0,951	0,964	0,842
		1	0,584	0,750	0,657	0,474	0,842
		Total	0,926	0,914	0,919	0,910	0,842
	BOVW-2000	**0**	0,968	0,933	0,951	0,964	0,842
		1	0,584	0,750	0,657	0,474	0,842
		Total	0,926	0,914	0,919	0,910	0,842
Random Forest	HOG	**0**	0,932	1	0,965	0,988	0,93
		1	0,999	0,411	0,582	0,803	0,93
		Total	0,94	0,935	0,923	0,967	0,93
	BOVW-500	**0**	0,93	1	0,964	0,988	0,929
		1	0,994	0,386	0,556	0,797	0,929
		Total	0,937	0,933	0,919	0,967	0,929
	BOVW-1000	**0**	0,929	1,000	0,963	0,988	0,929
		1	0,997	0,377	0,547	0,798	0,929
		Total	0,936	0,932	0,917	0,967	0,929
	BOVW-2000	**0**	0,928	1	0,963	0,987	0,929
		1	0,998	0,371	0,541	0,799	0,929
		Total	0,936	0,931	0,917	0,967	0,929

Table A.2. Test validation results on Tf-idf dataset

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,956	0,771	0,853	0,970	0,685
		1	0,113	0,450	0,181	0,128	0,685
		Total	0,904	0,751	0,812	0,918	0,685
	BOVW-500	**0**	0,969	0,686	0,803	0,978	0,761
		1	0,122	0,667	0,206	0,170	0,761
		Total	0,917	0,685	0,767	0,928	0,761
	BOVW-1000	**0**	0,958	0,673	0,790	0,957	0,760
		1	0,209	0,746	0,326	0,236	0,760
		Total	0,881	0,680	0,742	0,882	0,760
	BOVW-2000	**0**	0,962	0,647	0,774	0,958	0,630
		1	0,100	0,604	0,172	0,083	0,625
		Total	0,909	0,645	0,737	0,904	0,629
k-NN	HOG	**0**	0,936	0,853	0,893	0,936	0,479
		1	0,046	0,108	0,064	0,059	0,479
		Total	0,882	0,807	0,842	0,883	0,479
	BOVW-500	**0**	0,946	0,896	0,920	0,945	0,551
		1	0,119	0,216	0,153	0,084	0,551
		Total	0,895	0,854	0,873	0,892	0,551
	BOVW-1000	**0**	0,927	0,867	0,896	0,926	0,651
		1	0,264	0,411	0,321	0,190	0,651
		Total	0,858	0,820	0,837	0,849	0,651
	BOVW-2000	**0**	0,954	0,860	0,905	0,952	0,617
		1	0,147	0,369	0,210	0,097	0,617
		Total	0,905	0,830	0,862	0,900	0,617
Random Forest	HOG	**0**	0,939	1,000	0,968	0,971	0,707
		1	0	0	0	0,147	0,707
		Total	0	0,939	0	0,921	0,707
	BOVW-500	**0**	0,939	1,000	0,968	0,975	0,749
		1	0	0	0	0,200	0,749
		Total	0	0,939	0	0,927	0,749
	BOVW-1000	**0**	0,898	1,000	0,946	0,965	0,802
		1	1,000	0,020	0,040	0,439	0,802
		Total	0,909	0,898	0,852	0,911	0,802
	BOVW-2000	**0**	0,939	1,000	0,968	0,974	0,740
		1	0	0	0	0,183	0,740
		Total	0	0,939	0	0,926	0,740

Table A.3. 5-fold cross validation results on SEMILAR dataset

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,894	0,635	0,743	0,908	0,692
		1	0,272	0,644	0,382	0,311	0,692
		Total	0,786	0,637	0,680	0,804	0,692
	BOVW-500	**0**	0,899	0,625	0,737	0,905	0,702
		1	0,273	0,666	0,387	0,319	0,701
		Total	0,790	0,632	0,676	0,803	0,702
	BOVW-1000	**0**	0,892	0,624	0,734	0,889	0,667
		1	0,265	0,642	0,375	0,266	0,666
		Total	0,783	0,627	0,672	0,780	0,667
	BOVW-2000	**0**	0,879	0,607	0,718	0,870	0,617
		1	0,245	0,605	0,349	0,225	0,614
		Total	0,769	0,606	0,654	0,757	0,616
k-NN	HOG	**0**	0,937	0,891	0,914	0,927	0,803
		1	0,582	0,717	0,642	0,476	0,803
		Total	0,875	0,861	0,866	0,849	0,803
	BOVW-500	**0**	0,939	0,891	0,914	0,929	0,807
		1	0,583	0,724	0,646	0,480	0,807
		Total	0,877	0,862	0,867	0,851	0,807
	BOVW-1000	**0**	0,939	0,891	0,914	0,929	0,807
		1	0,583	0,724	0,646	0,480	0,807
		Total	0,877	0,862	0,867	0,851	0,807
	BOVW-2000	**0**	0,939	0,891	0,914	0,929	0,807
		1	0,583	0,724	0,646	0,480	0,807
		Total	0,877	0,862	0,867	0,851	0,807
Random Forest	HOG	**0**	0,887	0,999	0,940	0,973	0,906
		1	0,993	0,397	0,568	0,782	0,906
		Total	0,905	0,894	0,875	0,940	0,906
	BOVW-500	**0**	0,886	0,999	0,939	0,970	0,895
		1	0,983	0,390	0,558	0,764	0,895
		Total	0,903	0,892	0,872	0,934	0,895
	BOVW-1000	**0**	0,885	0,999	0,939	0,970	0,897
		1	0,988	0,384	0,553	0,768	0,897
		Total	0,903	0,892	0,871	0,935	0,897
	BOVW-2000	**0**	0,885	0,999	0,939	0,970	0,897
		1	0,988	0,384	0,553	0,768	0,897
		Total	0,903	0,892	0,871	0,935	0,897

Table A.4. Test validation results on SEMILAR dataset

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,903	0,777	0,835	0,938	0,702
		1	0,229	0,442	0,302	0,233	0,702
		Total	0,815	0,733	0,766	0,846	0,702
	BOVW-500	**0**	0,935	0,642	0,762	0,942	0,736
		1	0,228	0,704	0,344	0,285	0,736
		Total	0,843	0,650	0,707	0,856	0,736
	BOVW-1000	**0**	0,928	0,627	0,748	0,931	0,746
		1	0,294	0,760	0,424	0,317	0,746
		Total	0,820	0,650	0,693	0,827	0,746
	BOVW-2000	**0**	0,913	0,575	0,706	0,895	0,603
		1	0,183	0,633	0,283	0,164	0,603
		Total	0,818	0,583	0,651	0,800	0,603
k-NN	HOG	**0**	0,879	0,803	0,839	0,876	0,527
		1	0,166	0,261	0,203	0,139	0,527
		Total	0,786	0,732	0,756	0,780	0,527
	BOVW-500	**0**	0,876	0,833	0,854	0,874	0,520
		1	0,162	0,216	0,185	0,138	0,520
		Total	0,783	0,752	0,767	0,778	0,520
	BOVW-1000	**0**	0,858	0,794	0,824	0,852	0,574
		1	0,259	0,354	0,300	0,202	0,574
		Total	0,756	0,719	0,735	0,742	0,574
	BOVW-2000	**0**	0,886	0,846	0,866	0,884	0,563
		1	0,212	0,276	0,240	0,153	0,563
		Total	0,798	0,771	0,784	0,789	0,563
Random Forest	HOG	**0**	0,870	1,000	0,930	0,939	0,719
		1	0	0	0	0,287	0,719
		Total	0	0,870	0	0,854	0,719
	BOVW-500	**0**	0,872	0,999	0,931	0,947	0,761
		1	0,800	0,020	0,039	0,343	0,761
		Total	0,863	0,872	0,815	0,869	0,761
	BOVW-1000	**0**	0,834	0,999	0,909	0,939	0,781
		1	0,875	0,026	0,050	0,477	0,781
		Total	0,841	0,834	0,764	0,861	0,781
	BOVW-2000	**0**	0,870	1,000	0,930	0,940	0,730
		1	0	0	0	0,309	0,730
		Total	0	0,870	0	0,858	0,730

Table A.5. 5-fold cross validation results on Word2Vec dataset

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,907	0,600	0,722	0,916	0,733
		1	0,297	0,732	0,422	0,392	0,733
		Total	0,792	0,625	0,666	0,818	0,733
	BOVW-500	**0**	0,905	0,598	0,720	0,907	0,722
		1	0,295	0,729	0,420	0,348	0,721
		Total	0,791	0,623	0,664	0,802	0,722
	BOVW-1000	**0**	0,898	0,601	0,720	0,894	0,691
		1	0,289	0,704	0,410	0,296	0,689
		Total	0,784	0,620	0,662	0,782	0,691
	BOVW-2000	**0**	0,881	0,594	0,710	0,871	0,640
		1	0,271	0,653	0,383	0,254	0,636
		Total	0,767	0,605	0,648	0,756	0,639
k-NN	HOG	**0**	0,946	0,818	0,877	0,928	0,808
		1	0,503	0,798	0,617	0,444	0,808
		Total	0,863	0,814	0,829	0,837	0,808
	BOVW-500	**0**	0.965	0.842	0.899	0,949	0.854
		1	0.559	0.867	0.680	0,480	0.854
		Total	0.889	0.847	0.858	0,851	0.854
	BOVW-1000	**0**	0,965	0,842	0,899	0,945	0,854
		1	0,559	0,867	0,680	0,513	0,854
		Total	0,889	0,847	0,858	0,864	0,854
	BOVW-2000	**0**	0,965	0,842	0,899	0,946	0,854
		1	0,559	0,867	0,680	0,509	0,854
		Total	0,889	0,847	0,858	0,864	0,854
Random Forest	HOG	**0**	0,861	0,999	0,925	0,974	0,910
		1	0,990	0,303	0,464	0,775	0,910
		Total	0,885	0,869	0,839	0,936	0,910
	BOVW-500	**0**	0,877	0,998	0,934	0,977	0,921
		1	0,977	0,396	0,563	0,800	0,921
		Total	0,896	0,885	0,864	0,944	0,921
	BOVW-1000	**0**	0,877	0,999	0,934	0,979	0,927
		1	0,986	0,392	0,561	0,812	0,927
		Total	0,897	0,885	0,864	0,948	0,927
	BOVW-2000	**0**	0,877	0,999	0,934	0,978	0,925
		1	0,986	0,392	0,561	0,812	0,927
		Total	0,897	0,885	0,864	0,948	0,927

Table A.6. Test validation results on Word2Vec dataset

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,955	0,704	0,810	0,963	0,833
		1	0,322	0,811	0,461	0,445	0,831
		Total	0,862	0,719	0,759	0,886	0,832
	BOVW-500	**0**	0,966	0,651	0,778	0,956	0,826
		1	0,303	0,868	0,449	0,468	0,824
		Total	0,867	0,683	0,729	0,883	0,826
	BOVW-1000	**0**	0,929	0,635	0,754	0,916	0,772
		1	0,329	0,786	0,464	0,406	0,768
		Total	0,817	0,663	0,700	0,821	0,771
	BOVW-2000	**0**	0,936	0,626	0,750	0,921	0,691
		1	0,261	0,756	0,388	0,218	0,682
		Total	0,836	0,645	0,696	0,817	0,690
k-NN	HOG	**0**	0,865	0,675	0,758	0,860	0,534
		1	0,174	0,395	0,242	0,164	0,534
		Total	0,763	0,634	0,682	0,757	0,534
	BOVW-500	**0**	0,871	0,745	0,803	0,868	0,564
		1	0,202	0,371	0,262	0,181	0,564
		Total	0,772	0,689	0,723	0,766	0,564
	BOVW-1000	**0**	0,852	0,850	0,851	0,850	0,609
		1	0,350	0,354	0,352	0,268	0,609
		Total	0,759	0,758	0,759	0,742	0,609
	BOVW-2000	**0**	0,874	0,871	0,872	0,868	0,566
		1	0,274	0,279	0,276	0,188	0,566
		Total	0,784	0,783	0,784	0,767	0,566
Random Forest	HOG	**0**	0,852	0,999	0,920	0,948	0,777
		1	0,500	0,005	0,010	0,363	0,777
		Total	0,800	0,852	0,785	0,861	0,777
	BOVW-500	**0**	0,856	0,997	0,921	0,952	0,797
		1	0,700	0,036	0,068	0,425	0,797
		Total	0,832	0,854	0,794	0,874	0,797
	BOVW-1000	**0**	0,822	0,998	0,902	0,945	0,814
		1	0,867	0,051	0,096	0,527	0,814
		Total	0,830	0,822	0,752	0,867	0,814
	BOVW-2000	**0**	0,852	0,999	0,920	0,951	0,794
		1	0,500	0,005	0,010	0,387	0,794
		Total	0,800	0,851	0,784	0,867	0,794

Table A.7. 5-fold cross validation results on Tf-idf dataset with PCA

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,951	0,710	0,813	0,961	0,782
		1	0,230	0,704	0,347	0,399	0,782
		Total	0,872	0,710	0,762	0,899	0,782
	BOVW-500	**0**	0,951	0,711	0,814	0,961	0,782
		1	0,230	0,703	0,347	0,400	0,782
		Total	0,872	0,710	0,763	0,899	0,782
	BOVW-1000	**0**	0,951	0,711	0,814	0,961	0,782
		1	0,230	0,703	0,347	0,399	0,782
		Total	0,872	0,710	0,763	0,899	0,782
	BOVW-2000	**0**	0,951	0,709	0,812	0,960	0,781
		1	0,229	0,704	0,346	0,397	0,781
		Total	0,872	0,708	0,761	0,898	0,781
k-NN	HOG	**0**	0,968	0,934	0,951	0,964	0,842
		1	0,584	0,750	0,657	0,474	0,842
		Total	0,926	0,914	0,919	0,911	0,842
	BOVW-500	**0**	0,968	0,934	0,951	0,964	0,842
		1	0,584	0,750	0,656	0,474	0,842
		Total	0,926	0,914	0,919	0,910	0,842
	BOVW-1000	**0**	0,968	0,934	0,951	0,964	0,842
		1	0,584	0,750	0,657	0,474	0,842
		Total	0,926	0,914	0,919	0,910	0,842
	BOVW-2000	**0**	0,968	0,934	0,951	0,964	0,842
		1	0,584	0,750	0,657	0,474	0,842
		Total	0,926	0,914	0,919	0,911	0,842
Random Forest	HOG	**0**	0,931	1,000	0,964	0,988	0,929
		1	0,992	0,396	0,566	0,797	0,929
		Total	0,938	0,934	0,920	0,967	0,929
	BOVW-500	**0**	0,931	1,000	0,964	0,987	0,929
		1	0,993	0,398	0,568	0,797	0,929
		Total	0,938	0,934	0,921	0,966	0,929
	BOVW-1000	**0**	0,931	1,000	0,964	0,987	0,929
		1	0,993	0,399	0,569	0,798	0,929
		Total	0,938	0,934	0,921	0,967	0,929
	BOVW-2000	**0**	0,931	1,000	0,964	0,987	0,928
		1	0,993	0,399	0,569	0,797	0,928
		Total	0,938	0,934	0,921	0,966	0,928

Table A.8. Test validation results on Tf-idf dataset with PCA

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,965	0,700	0,811	0,972	0,826
		1	0,231	0,782	0,357	0,438	0,827
		Total	0,889	0,708	0,764	0,917	0,826
	BOVW-500	**0**	0,964	0,696	0,808	0,972	0,827
		1	0,228	0,777	0,353	0,441	0,827
		Total	0,888	0,704	0,761	0,917	0,827
	BOVW-1000	**0**	0,964	0,700	0,811	0,972	0,827
		1	0,230	0,777	0,355	0,441	0,827
		Total	0,888	0,708	0,764	0,917	0,827
	BOVW-2000	**0**	0,963	0,697	0,809	0,972	0,827
		1	0,227	0,772	0,351	0,441	0,827
		Total	0,887	0,704	0,761	0,917	0,827
k-NN	HOG	**0**	0,902	0,901	0,902	0,905	0,543
		1	0,152	0,152	0,152	0,114	0,543
		Total	0,824	0,824	0,824	0,823	0,543
	BOVW-500	**0**	0,923	0,903	0,913	0,926	0,651
		1	0,294	0,350	0,319	0,194	0,651
		Total	0,858	0,845	0,851	0,850	0,651
	BOVW-1000	**0**	0,923	0,886	0,904	0,925	0,644
		1	0,268	0,360	0,307	0,189	0,644
		Total	0,855	0,832	0,842	0,848	0,644
	BOVW-2000	**0**	0,924	0,884	0,904	0,923	0,638
		1	0,272	0,376	0,316	0,196	0,638
		Total	0,857	0,831	0,843	0,848	0,638
Random Forest	HOG	**0**	0,902	1,000	0,948	0,968	0,816
		1	1,000	0,056	0,106	0,466	0,816
		Total	0,912	0,902	0,861	0,916	0,816
	BOVW-500	**0**	0,902	1,000	0,948	0,967	0,813
		1	1,000	0,056	0,106	0,485	0,813
		Total	0,912	0,902	0,861	0,917	0,813
	BOVW-1000	**0**	0,901	1,000	0,948	0,967	0,812
		1	1,000	0,046	0,087	0,480	0,812
		Total	0,911	0,901	0,859	0,916	0,812
	BOVW-2000	**0**	0,901	0,999	0,948	0,970	0,826
		1	0,909	0,051	0,096	0,478	0,826
		Total	0,902	0,901	0,859	0,919	0,826

Table A.9. 5-fold cross validation results on SEMILAR dataset with PCA

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,901	0,626	0,739	0,914	0,718
		1	0,275	0,674	0,391	0,403	0,718
		Total	0,792	0,634	0,678	0,825	0,718
	BOVW-500	**0**	0,901	0,626	0,739	0,914	0,718
		1	0,275	0,674	0,391	0,402	0,718
		Total	0,792	0,634	0,678	0,825	0,718
	BOVW-1000	**0**	0,901	0,626	0,739	0,914	0,717
		1	0,275	0,673	0,391	0,402	0,717
		Total	0,792	0,634	0,678	0,824	0,717
	BOVW-2000	**0**	0,901	0,624	0,737	0,913	0,716
		1	0,274	0,674	0,390	0,401	0,716
		Total	0,792	0,633	0,677	0,824	0,716
k-NN	HOG	**0**	0,937	0,891	0,914	0,927	0,803
		1	0,582	0,717	0,643	0,477	0,803
		Total	0,875	0,861	0,866	0,849	0,803
	BOVW-500	**0**	0,939	0,891	0,914	0,929	0,807
		1	0,583	0,724	0,646	0,480	0,807
		Total	0,877	0,862	0,867	0,851	0,807
	BOVW-1000	**0**	0,939	0,891	0,914	0,929	0,807
		1	0,583	0,724	0,646	0,480	0,807
		Total	0,877	0,862	0,867	0,851	0,807
	BOVW-2000	**0**	0,939	0,891	0,914	0,929	0,807
		1	0,583	0,724	0,646	0,480	0,807
		Total	0,877	0,862	0,867	0,851	0,807
Random Forest	HOG	**0**	0,886	0,998	0,939	0,970	0,894
		1	0,977	0,395	0,562	0,763	0,894
		Total	0,902	0,893	0,873	0,933	0,894
	BOVW-500	**0**	0,888	0,998	0,940	0,970	0,895
		1	0,981	0,400	0,569	0,766	0,895
		Total	0,904	0,894	0,875	0,934	0,895
	BOVW-1000	**0**	0,888	0,998	0,940	0,969	0,894
		1	0,980	0,401	0,569	0,764	0,894
		Total	0,904	0,894	0,875	0,934	0,894
	BOVW-2000	**0**	0,888	0,998	0,940	0,969	0,894
		1	0,978	0,402	0,569	0,764	0,894
		Total	0,903	0,894	0,875	0,934	0,894

Table A.10. Test validation results on SEMILAR dataset with PCA

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,942	0,654	0,772	0,943	0,799
		1	0,322	0,804	0,459	0,479	0,799
		Total	0,837	0,679	0,719	0,864	0,799
	BOVW-500	**0**	0,943	0,655	0,773	0,944	0,799
		1	0,322	0,804	0,460	0,476	0,799
		Total	0,837	0,680	0,720	0,865	0,799
	BOVW-1000	**0**	0,943	0,657	0,774	0,944	0,799
		1	0,323	0,804	0,461	0,479	0,799
		Total	0,838	0,682	0,721	0,865	0,799
	BOVW-2000	**0**	0,942	0,648	0,768	0,943	0,800
		1	0,318	0,804	0,456	0,481	0,800
		Total	0,836	0,675	0,715	0,865	0,800
k-NN	HOG	**0**	0,840	0,751	0,793	0,842	0,536
		1	0,197	0,299	0,237	0,182	0,536
		Total	0,731	0,674	0,699	0,730	0,536
	BOVW-500	**0**	0,848	0,813	0,830	0,845	0,549
		1	0,239	0,288	0,261	0,195	0,549
		Total	0,745	0,724	0,734	0,735	0,549
	BOVW-1000	**0**	0,862	0,809	0,834	0,857	0,588
		1	0,280	0,365	0,317	0,211	0,588
		Total	0,763	0,734	0,747	0,747	0,588
	BOVW-2000	**0**	0,855	0,846	0,850	0,851	0,572
		1	0,281	0,295	0,288	0,204	0,572
		Total	0,757	0,752	0,755	0,742	0,572
Random Forest	HOG	**0**	0,838	1,000	0,912	0,941	0,797
		1	1,000	0,055	0,105	0,542	0,797
		Total	0,866	0,840	0,775	0,873	0,797
	BOVW-500	**0**	0,840	0,999	0,913	0,939	0,787
		1	0,950	0,070	0,131	0,505	0,787
		Total	0,859	0,842	0,780	0,865	0,787
	BOVW-1000	**0**	0,840	0,998	0,912	0,935	0,789
		1	0,900	0,066	0,124	0,526	0,789
		Total	0,850	0,841	0,779	0,866	0,789
	BOVW-2000	**0**	0,839	0,998	0,912	0,942	0,802
		1	0,895	0,063	0,117	0,528	0,802
		Total	0,849	0,840	0,777	0,872	0,802

Table A.11. 5-fold cross validation results on Word2Vec dataset with PCA

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,911	0,594	0,719	0,918	0,742
		1	0,298	0,747	0,426	0,424	0,742
		Total	0,796	0,623	0,664	0,825	0,742
	BOVW-500	**0**	0,910	0,595	0,720	0,917	0,742
		1	0,298	0,746	0,426	0,424	0,742
		Total	0,796	0,623	0,665	0,825	0,742
	BOVW-1000	**0**	0,910	0,595	0,720	0,917	0,742
		1	0,298	0,746	0,426	0,424	0,742
		Total	0,796	0,623	0,665	0,825	0,742
	BOVW-2000	**0**	0,910	0,594	0,719	0,917	0,741
		1	0,298	0,746	0,426	0,424	0,741
		Total	0,795	0,623	0,664	0,825	0,741
k-NN	HOG	**0**	0,965	0,842	0,899	0,946	0,855
		1	0,558	0,868	0,679	0,514	0,855
		Total	0,889	0,847	0,858	0,865	0,855
	BOVW-500	**0**	0,965	0,842	0,900	0,945	0,854
		1	0,559	0,867	0,680	0,513	0,854
		Total	0,889	0,847	0,858	0,864	0,854
	BOVW-1000	**0**	0,965	0,842	0,900	0,945	0,854
		1	0,559	0,867	0,680	0,513	0,854
		Total	0,889	0,847	0,858	0,864	0,854
	BOVW-2000	**0**	0,965	0,842	0,900	0,945	0,854
		1	0,559	0,867	0,680	0,513	0,854
		Total	0,889	0,847	0,858	0,864	0,854
Random Forest	HOG	**0**	0,880	0,997	0,935	0,978	0,923
		1	0,972	0,409	0,576	0,804	0,923
		Total	0,897	0,887	0,868	0,945	0,923
	BOVW-500	**0**	0,879	0,997	0,935	0,977	0,921
		1	0,971	0,407	0,574	0,801	0,921
		Total	0,897	0,887	0,867	0,944	0,921
	BOVW-1000	**0**	0,879	0,997	0,935	0,977	0,920
		1	0,972	0,407	0,574	0,801	0,920
		Total	0,897	0,887	0,867	0,944	0,920
	BOVW-2000	**0**	0,879	0,997	0,935	0,977	0,919
		1	0,973	0,406	0,573	0,799	0,919
		Total	0,897	0,887	0,867	0,943	0,919

Table A.12. Test validation results on Word2Vec dataset with PCA

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,962	0,627	0,759	0,954	0,844
		1	0,351	0,891	0,504	0,550	0,843
		Total	0,849	0,676	0,712	0,879	0,844
	BOVW-500	**0**	0,955	0,635	0,763	0,952	0,838
		1	0,351	0,868	0,500	0,545	0,838
		Total	0,843	0,678	0,714	0,876	0,838
	BOVW-1000	**0**	0,955	0,635	0,763	0,952	0,838
		1	0,351	0,868	0,500	0,547	0,837
		Total	0,843	0,678	0,714	0,876	0,837
	BOVW-2000	**0**	0,957	0,629	0,759	0,951	0,838
		1	0,349	0,875	0,499	0,546	0,837
		Total	0,844	0,674	0,711	0,876	0,838
k-NN	HOG	**0**	0,838	0,724	0,777	0,836	0,561
		1	0,239	0,383	0,295	0,209	0,561
		Total	0,727	0,661	0,687	0,720	0,561
	BOVW-500	**0**	0,843	0,847	0,845	0,847	0,597
		1	0,313	0,307	0,310	0,242	0,597
		Total	0,745	0,747	0,746	0,735	0,597
	BOVW-1000	**0**	0,851	0,823	0,837	0,849	0,608
		1	0,322	0,370	0,344	0,275	0,608
		Total	0,753	0,739	0,745	0,743	0,608
	BOVW-2000	**0**	0,854	0,840	0,847	0,851	0,613
		1	0,344	0,370	0,356	0,274	0,613
		Total	0,759	0,752	0,756	0,744	0,613
Random Forest	HOG	**0**	0,823	0,997	0,902	0,952	0,835
		1	0,824	0,056	0,106	0,539	0,835
		Total	0,823	0,823	0,755	0,876	0,835
	BOVW-500	**0**	0,823	0,996	0,901	0,951	0,829
		1	0,762	0,062	0,115	0,542	0,829
		Total	0,812	0,822	0,755	0,875	0,829
	BOVW-1000	**0**	0,826	0,998	0,904	0,947	0,817
		1	0,909	0,078	0,143	0,533	0,817
		Total	0,842	0,827	0,763	0,870	0,817
	BOVW-2000	**0**	0,825	0,996	0,902	0,950	0,822
		1	0,818	0,070	0,129	0,515	0,822
		Total	0,823	0,825	0,759	0,869	0,822

Table A.13. 5-fold cross validation results on Tf-idf dataset with new split

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,943	0,670	0,784	0,951	0,736
		1	0,201	0,673	0,309	0,296	0,737
		Total	0,862	0,671	0,732	0,880	0,736
	BOVW-500	**0**	0,948	0,694	0,802	0,953	0,753
		1	0,218	0,692	0,331	0,272	0,753
		Total	0,868	0,694	0,75	0,878	0,753
	BOVW-1000	**0**	0,941	0,668	0,782	0,940	0,703
		1	0,197	0,662	0,304	0,198	0,701
		Total	0,860	0,668	0,729	0,859	0,702
	BOVW-2000	**0**	0,929	0,613	0,739	0,923	0,627
		1	0,165	0,620	0,260	0,148	0,623
		Total	0,845	0,614	0,686	0,838	0,626
k-NN	HOG	**0**	0,965	0,928	0,946	0,961	0,829
		1	0,555	0,725	0,628	0,439	0,829
		Total	0,920	0,906	0,911	0,904	0,829
	BOVW-500	**0**	0,965	0,928	0,946	0,961	0,829
		1	0,555	0,725	0,628	0,439	0,829
		Total	0,920	0,906	0,911	0,904	0,829
	BOVW-1000	**0**	0,965	0,928	0,946	0,961	0,829
		1	0,555	0,725	0,628	0,439	0,829
		Total	0,920	0,906	0,911	0,904	0,829
	BOVW-2000	**0**	0,965	0,928	0,946	0,961	0,829
		1	0,555	0,725	0,628	0,439	0,829
		Total	0,920	0,906	0,911	0,904	0,829
Random Forest	HOG	**0**	0,927	1,000	0,962	0,985	0,917
		1	0,998	0,358	0,527	0,774	0,917
		Total	0,935	0,930	0,914	0,962	0,917
	BOVW-500	**0**	0,924	1,000	0,960	0,985	0,917
		1	0,328	0,493	0,548	0,917	0,768
		Total	0,931	0,926	0,909	0,961	0,917
	BOVW-1000	**0**	0,923	1,000	0,960	0,985	0,918
		1	0,996	0,322	0,487	0,770	0,918
		Total	0,931	0,926	0,908	0,962	0,918
	BOVW-2000	**0**	0,923	1,000	0,960	0,985	0,917
		1	0,998	0,319	0,483	0,773	0,917
		Total	0,931	0,925	0,908	0,962	0,917

Table A.14. Test validation results on Tf-idf dataset with new split

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,943	0,667	0,781	0,949	0,729
		1	0,198	0,671	0,306	0,291	0,729
		Total	0,862	0,667	0,729	0,877	0,729
	BOVW-500	**0**	0,946	0,698	0,803	0,951	0,748
		1	0,215	0,676	0,326	0,287	0,747
		Total	0,866	0,695	0,751	0,878	0,748
	BOVW-1000	**0**	0,941	0,663	0,778	0,940	0,701
		1	0,194	0,664	0,300	0,199	0,700
		Total	0,860	0,663	0,726	0,859	0,701
	BOVW-2000	**0**	0,929	0,600	0,729	0,923	0,627
		1	0,160	0,623	0,255	0,149	0,627
		Total	0,845	0,603	0,677	0,839	0,627
k-NN	HOG	**0**	0,934	0,918	0,926	0,928	0,680
		1	0,412	0,469	0,439	0,243	0,680
		Total	0,877	0,869	0,873	0,853	0,680
	BOVW-500	**0**	0,938	0,921	0,930	0,929	0,680
		1	0,440	0,505	0,470	0,257	0,680
		Total	0,884	0,876	0,880	0,856	0,680
	BOVW-1000	**0**	0,942	0,904	0,923	0,925	0,661
		1	0,411	0,550	0,471	0,241	0,661
		Total	0,885	0,865	0,873	0,851	0,661
	BOVW-2000	**0**	0,938	0,921	0,930	0,929	0,679
		1	0,440	0,505	0,470	0,250	0,679
		Total	0,884	0,876	0,880	0,856	0,679
Random Forest	HOG	**0**	0,923	1,000	0,960	0,981	0,889
		1	0,998	0,319	0,483	0,672	0,889
		Total	0,931	0,926	0,908	0,948	0,889
	BOVW-500	**0**	0,925	1,000	0,961	0,982	0,899
		1	0,995	0,336	0,503	0,707	0,899
		Total	0,932	0,927	0,911	0,952	0,899
	BOVW-1000	**0**	0,924	1,000	0,961	0,981	0,893
		1	0,998	0,330	0,496	0,700	0,893
		Total	0,932	0,927	0,910	0,950	0,893
	BOVW-2000	**0**	0,923	1,000	0,960	0,980	0,887
		1	0,998	0,323	0,488	0,689	0,887
		Total	0,932	0,926	0,909	0,948	0,887

Table A.15. 5-fold cross validation results on SEMILAR dataset with new split

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,895	0,635	0,743	0,908	0,694
		1	0,272	0,647	0,383	0,315	0,694
		Total	0,787	0,637	0,680	0,805	0,694
	BOVW-500	**0**	0,899	0,626	0,738	0,904	0,701
		1	0,272	0,665	0,386	0,319	0,700
		Total	0,790	0,633	0,677	0,802	0,700
	BOVW-1000	**0**	0,892	0,620	0,732	0,889	0,667
		1	0,263	0,645	0,374	0,267	0,666
		Total	0,783	0,624	0,669	0,781	0,667
	BOVW-2000	**0**	0,878	0,605	0,716	0,869	0,615
		1	0,243	0,603	0,346	0,223	0,612
		Total	0,768	0,604	0,652	0,757	0,614
k-NN	HOG	**0**	0,933	0,881	0,906	0,921	0,790
		1	0,553	0,700	0,618	0,441	0,790
		Total	0,867	0,849	0,856	0,837	0,790
	BOVW-500	**0**	0,934	0,883	0,908	0,924	0,793
		1	0,559	0,702	0,622	0,454	0,793
		Total	0,868	0,852	0,858	0,842	0,793
	BOVW-1000	**0**	0,965	0,928	0,946	0,961	0,829
		1	0,555	0,725	0,628	0,439	0,829
		Total	0,920	0,906	0,911	0,904	0,829
	BOVW-2000	**0**	0,965	0,928	0,946	0,961	0,829
		1	0,555	0,725	0,628	0,439	0,829
		Total	0,920	0,906	0,911	0,904	0,829
Random Forest	HOG	**0**	0,880	0,999	0,936	0,969	0,892
		1	0,989	0,353	0,520	0,755	0,892
		Total	0,899	0,887	0,863	0,932	0,892
	BOVW-500	**0**	0,877	0,998	0,934	0,965	0,879
		1	0,978	0,335	0,499	0,732	0,879
		Total	0,895	0,883	0,858	0,924	0,879
	BOVW-1000	**0**	0,876	0,999	0,933	0,965	0,880
		1	0,983	0,329	0,493	0,735	0,880
		Total	0,895	0,882	0,857	0,925	0,880
	BOVW-2000	**0**	0,875	0,999	0,933	0,966	0,917
		1	0,986	0,325	0,489	0,741	0,917
		Total	0,895	0,882	0,856	0,927	0,917

Table A.16. Test validation results on SEMILAR dataset with new split

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,893	0,633	0,741	0,907	0,692
		1	0,273	0,646	0,383	0,311	0,692
		Total	0,784	0,635	0,678	0,802	0,692
	BOVW-500	**0**	0,897	0,620	0,734	0,900	0,697
		1	0,272	0,667	0,386	0,323	0,696
		Total	0,788	0,628	0,673	0,799	0,697
	BOVW-1000	**0**	0,892	0,606	0,721	0,886	0,667
		1	0,261	0,656	0,374	0,265	0,666
		Total	0,781	0,614	0,660	0,777	0,667
	BOVW-2000	**0**	0,880	0,593	0,709	0,871	0,623
		1	0,245	0,619	0,351	0,232	0,621
		Total	0,768	0,598	0,646	0,759	0,623
k-NN	HOG	**0**	0,895	0,868	0,881	0,883	0,683
		1	0,457	0,521	0,487	0,317	0,683
		Total	0,818	0,807	0,812	0,783	0,683
	BOVW-500	**0**	0,892	0,886	0,889	0,883	0,685
		1	0,479	0,494	0,486	0,324	0,685
		Total	0,819	0,817	0,818	0,785	0,685
	BOVW-1000	**0**	0,895	0,882	0,888	0,887	0,697
		1	0,481	0,515	0,498	0,336	0,697
		Total	0,823	0,817	0,820	0,790	0,697
	BOVW-2000	**0**	0,895	0,882	0,888	0,887	0,697
		1	0,476	0,510	0,492	0,336	0,697
		Total	0,823	0,817	0,820	0,790	0,697
Random Forest	HOG	**0**	0,873	1,000	0,932	0,962	0,864
		1	0,997	0,315	0,479	0,693	0,864
		Total	0,894	0,880	0,852	0,915	0,864
	BOVW-500	**0**	0,877	0,999	0,934	0,962	0,869
		1	0,989	0,342	0,509	0,714	0,869
		Total	0,897	0,884	0,860	0,919	0,869
	BOVW-1000	**0**	0,876	0,999	0,934	0,961	0,880
		1	0,993	0,334	0,500	0,705	0,880
		Total	0,896	0,883	0,858	0,916	0,880
	BOVW-2000	**0**	0,875	1,000	0,933	0,961	0,864
		1	0,994	0,327	0,492	0,704	0,864
		Total	0,896	0,882	0,856	0,916	0,864

Table A.17. 5-fold cross validation results on Word2Vec dataset with new split

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,908	0,602	0,724	0,917	0,737
		1	0,299	0,735	0,425	0,401	0,738
		Total	0,794	0,627	0,668	0,820	0,737
	BOVW-500	**0**	0,907	0,601	0,723	0,908	0,726
		1	0,297	0,732	0,423	0,352	0,725
		Total	0,793	0,626	0,667	0,804	0,726
	BOVW-1000	**0**	0,897	0,603	0,721	0,892	0,688
		1	0,289	0,699	0,409	0,294	0,686
		Total	0,783	0,621	0,663	0,780	0,688
	BOVW-2000	**0**	0,881	0,597	0,712	0,871	0,639
		1	0,271	0,650	0,383	0,251	0,634
		Total	0,767	0,607	0,650	0,755	0,638
k-NN	HOG	**0**	0,962	0,829	0,891	0,941	0,843
		1	0,536	0,857	0,660	0,491	0,843
		Total	0,882	0,835	0,847	0,857	0,843
	BOVW-500	**0**	0,962	0,828	0,890	0,940	0,842
		1	0,535	0,857	0,658	0,486	0,842
		Total	0,882	0,833	0,847	0,855	0,842
	BOVW-1000	**0**	0,962	0,828	0,890	0,940	0,842
		1	0,535	0,857	0,658	0,486	0,842
		Total	0,882	0,833	0,847	0,855	0,842
	BOVW-2000	**0**	0,962	0,828	0,890	0,940	0,842
		1	0,535	0,857	0,658	0,486	0,842
		Total	0,882	0,833	0,847	0,855	0,842
Random Forest	HOG	**0**	0,875	0,999	0,933	0,982	0,936
		1	0,992	0,381	0,551	0,831	0,936
		Total	0,897	0,884	0,862	0,954	0,936
	BOVW-500	**0**	0,868	0,998	0,929	0,974	0,911
		1	0,971	0,343	0,507	0,776	0,911
		Total	0,888	0,875	0,850	0,937	0,911
	BOVW-1000	**0**	0,868	0,999	0,929	0,976	0,916
		1	0,982	0,339	0,504	0,787	0,916
		Total	0,889	0,875	0,849	0,940	0,916
	BOVW-2000	**0**	0,867	0,999	0,928	0,977	0,921
		1	0,986	0,335	0,500	0,797	0,921
		Total	0,889	0,875	0,848	0,944	0,921

Table A.18. Test validation results on Word2Vec dataset with new split

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,906	0,597	0,720	0,913	0,728
		1	0,296	0,732	0,422	0,395	0,729
		Total	0,791	0,622	0,664	0,815	0,728
	BOVW-500	**0**	0,903	0,596	0,718	0,904	0,717
		1	0,293	0,725	0,418	0,348	0,716
		Total	0,789	0,620	0,662	0,799	0,717
	BOVW-1000	**0**	0,894	0,602	0,719	0,892	0,686
		1	0,287	0,690	0,405	0,293	0,684
		Total	0,779	0,619	0,660	0,779	0,685
	BOVW-2000	**0**	0,871	0,577	0,694	0,859	0,615
		1	0,257	0,632	0,365	0,238	0,611
		Total	0,756	0,587	0,632	0,743	0,614
k-NN	HOG	**0**	0,889	0,842	0,865	0,873	0,679
		1	0,444	0,545	0,489	0,321	0,679
		Total	0,805	0,786	0,794	0,769	0,679
	BOVW-500	**0**	0,890	0,844	0,866	0,862	0,651
		1	0,449	0,552	0,495	0,324	0,651
		Total	0,807	0,789	0,797	0,761	0,651
	BOVW-1000	**0**	0,888	0,887	0,887	0,854	0,623
		1	0,514	0,518	0,516	0,328	0,623
		Total	0,818	0,817	0,818	0,755	0,623
	BOVW-2000	**0**	0,889	0,888	0,888	0,850	0,620
		1	0,451	0,512	0,479	0,323	0,620
		Total	0,805	0,783	0,794	0,754	0,620
Random Forest	HOG	**0**	0,862	1,000	0,926	0,963	0,873
		1	0,997	0,307	0,470	0,705	0,873
		Total	0,887	0,869	0,840	0,914	0,873
	BOVW-500	**0**	0,868	0,998	0,928	0,965	0,883
		1	0,979	0,343	0,508	0,730	0,883
		Total	0,889	0,875	0,849	0,921	0,883
	BOVW-1000	**0**	0,866	0,999	0,928	0,964	0,916
		1	0,991	0,333	0,499	0,726	0,916
		Total	0,890	0,874	0,847	0,919	0,916
	BOVW-2000	**0**	0,864	0,999	0,927	0,963	0,877
		1	0,987	0,321	0,485	0,721	0,877
		Total	0,887	0,872	0,843	0,917	0,877

Table A.19. 5-fold cross validation results on Tf-idf dataset with PCA with new split

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,951	0,708	0,812	0,961	0,782
		1	0,230	0,706	0,346	0,398	0,782
		Total	0,872	0,708	0,761	0,899	0,782
	BOVW-500	**0**	0,951	0,710	0,813	0,961	0,783
		1	0,230	0,705	0,347	0,399	0,783
		Total	0,872	0,709	0,762	0,899	0,783
	BOVW-1000	**0**	0,951	0,709	0,813	0,961	0,783
		1	0,230	0,706	0,347	0,398	0,783
		Total	0,872	0,709	0,762	0,899	0,783
	BOVW-2000	**0**	0,951	0,707	0,811	0,960	0,781
		1	0,229	0,705	0,345	0,396	0,781
		Total	0,872	0,707	0,760	0,899	0,781
k-NN	HOG	**0**	0,965	0,928	0,946	0,961	0,829
		1	0,555	0,725	0,629	0,439	0,829
		Total	0,920	0,906	0,911	0,904	0,829
	BOVW-500	**0**	0,965	0,928	0,946	0,961	0,829
		1	0,555	0,725	0,628	0,439	0,829
		Total	0,920	0,906	0,911	0,904	0,829
	BOVW-1000	**0**	0,965	0,928	0,946	0,961	0,829
		1	0,555	0,725	0,628	0,439	0,829
		Total	0,920	0,906	0,911	0,904	0,829
	BOVW-2000	**0**	0,965	0,928	0,946	0,961	0,829
		1	0,555	0,725	0,628	0,439	0,829
		Total	0,920	0,906	0,911	0,904	0,829
Random Forest	HOG	**0**	0,925	1,000	0,961	0,985	0,918
		1	0,992	0,345	0,512	0,771	0,918
		Total	0,933	0,928	0,912	0,962	0,918
	BOVW-500	**0**	0,925	1,000	0,961	0,985	0,918
		1	0,992	0,345	0,512	0,769	0,918
		Total	0,933	0,928	0,912	0,962	0,918
	BOVW-1000	**0**	0,925	1,000	0,961	0,985	0,916
		1	0,989	0,346	0,513	0,769	0,916
		Total	0,933	0,928	0,912	0,961	0,916
	BOVW-2000	**0**	0,926	1,000	0,961	0,985	0,916
		1	0,990	0,348	0,515	0,770	0,916
		Total	0,933	0,928	0,912	0,961	0,916

Table A.20. Test validation results on Tf-idf dataset with PCA with new split

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,949	0,709	0,812	0,958	0,772
		1	0,225	0,690	0,340	0,392	0,772
		Total	0,870	0,707	0,760	0,896	0,772
	BOVW-500	**0**	0,950	0,712	0,814	0,958	0,773
		1	0,228	0,693	0,343	0,394	0,773
		Total	0,871	0,710	0,762	0,896	0,773
	BOVW-1000	**0**	0,949	0,712	0,813	0,958	0,772
		1	0,227	0,691	0,342	0,393	0,772
		Total	0,871	0,709	0,762	0,896	0,772
	BOVW-2000	**0**	0,949	0,709	0,812	0,957	0,770
		1	0,225	0,692	0,340	0,391	0,770
		Total	0,870	0,707	0,760	0,895	0,770
k-NN	HOG	**0**	0,936	0,911	0,923	0,928	0,679
		1	0,402	0,489	0,441	0,245	0,679
		Total	0,877	0,865	0,871	0,853	0,679
	BOVW-500	**0**	0,941	0,914	0,928	0,930	0,685
		1	0,433	0,536	0,479	0,258	0,685
		Total	0,886	0,873	0,879	0,857	0,685
	BOVW-1000	**0**	0,942	0,910	0,926	0,930	0,682
		1	0,426	0,543	0,477	0,254	0,682
		Total	0,886	0,870	0,877	0,856	0,682
	BOVW-2000	**0**	0,940	0,918	0,929	0,929	0,680
		1	0,438	0,525	0,477	0,257	0,680
		Total	0,886	0,875	0,880	0,856	0,680
Random Forest	HOG	**0**	0,926	1,000	0,961	0,983	0,901
		1	0,997	0,346	0,514	0,714	0,901
		Total	0,934	0,929	0,913	0,953	0,901
	BOVW-500	**0**	0,926	1,000	0,961	0,982	0,900
		1	0,995	0,345	0,512	0,712	0,900
		Total	0,933	0,928	0,912	0,953	0,900
	BOVW-1000	**0**	0,926	1,000	0,962	0,981	0,896
		1	0,993	0,349	0,516	0,709	0,896
		Total	0,933	0,929	0,913	0,952	0,896
	BOVW-2000	**0**	0,926	1,000	0,961	0,981	0,896
		1	0,994	0,348	0,515	0,708	0,896
		Total	0,933	0,929	0,913	0,951	0,896

Table A.21. 5-fold cross validation results on SEMILAR dataset with PCA with new split

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,901	0,626	0,739	0,914	0,718
		1	0,275	0,675	0,391	0,401	0,718
		Total	0,792	0,634	0,678	0,825	0,718
	BOVW-500	**0**	0,902	0,626	0,739	0,914	0,718
		1	0,275	0,676	0,391	0,400	0,718
		Total	0,793	0,634	0,678	0,825	0,718
	BOVW-1000	**0**	0,902	0,626	0,739	0,914	0,718
		1	0,276	0,676	0,391	0,400	0,718
		Total	0,793	0,635	0,679	0,825	0,718
	BOVW-2000	**0**	0,902	0,624	0,738	0,913	0,717
		1	0,275	0,677	0,391	0,400	0,717
		Total	0,793	0,633	0,677	0,824	0,717
k-NN	HOG	**0**	0,933	0,881	0,906	0,921	0,790
		1	0,553	0,700	0,618	0,441	0,790
		Total	0,867	0,849	0,856	0,837	0,790
	BOVW-500	**0**	0,934	0,883	0,908	0,924	0,793
		1	0,559	0,702	0,622	0,454	0,793
		Total	0,868	0,852	0,858	0,842	0,793
	BOVW-1000	**0**	0,934	0,883	0,908	0,924	0,793
		1	0,559	0,702	0,622	0,454	0,793
		Total	0,868	0,852	0,858	0,842	0,793
	BOVW-2000	**0**	0,934	0,883	0,908	0,924	0,793
		1	0,559	0,702	0,622	0,454	0,793
		Total	0,868	0,852	0,858	0,842	0,793
Random Forest	HOG	**0**	0,879	0,998	0,935	0,965	0,879
		1	0,973	0,349	0,513	0,735	0,879
		Total	0,895	0,885	0,861	0,925	0,879
	BOVW-500	**0**	0,879	0,998	0,935	0,965	0,879
		1	0,973	0,348	0,513	0,734	0,879
		Total	0,895	0,885	0,861	0,925	0,879
	BOVW-1000	**0**	0,879	0,998	0,935	0,964	0,878
		1	0,975	0,350	0,515	0,733	0,878
		Total	0,896	0,885	0,862	0,924	0,878
	BOVW-2000	**0**	0,879	0,998	0,935	0,964	0,877
		1	0,973	0,345	0,510	0,732	0,877
		Total	0,895	0,884	0,861	0,923	0,877

Table A.22. Test validation results on SEMILAR dataset with PCA with new split

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,898	0,621	0,734	0,912	0,714
		1	0,273	0,670	0,388	0,407	0,714
		Total	0,789	0,629	0,673	0,823	0,714
	BOVW-500	**0**	0,898	0,622	0,735	0,912	0,713
		1	0,274	0,669	0,388	0,406	0,713
		Total	0,789	0,631	0,675	0,823	0,713
	BOVW-1000	**0**	0,898	0,623	0,735	0,911	0,713
		1	0,273	0,668	0,388	0,407	0,713
		Total	0,789	0,631	0,674	0,823	0,713
	BOVW-2000	**0**	0,898	0,621	0,734	0,910	0,712
		1	0,272	0,667	0,387	0,405	0,712
		Total	0,788	0,629	0,673	0,822	0,712
k-NN	HOG	**0**	0,896	0,852	0,873	0,882	0,679
		1	0,436	0,538	0,481	0,308	0,679
		Total	0,815	0,796	0,805	0,781	0,679
	BOVW-500	**0**	0,895	0,879	0,887	0,886	0,694
		1	0,476	0,515	0,495	0,331	0,694
		Total	0,822	0,815	0,818	0,789	0,694
	BOVW-1000	**0**	0,895	0,869	0,882	0,886	0,696
		1	0,459	0,523	0,489	0,326	0,696
		Total	0,819	0,808	0,813	0,788	0,696
	BOVW-2000	**0**	0,893	0,902	0,897	0,886	0,695
		1	0,514	0,490	0,502	0,344	0,695
		Total	0,826	0,829	0,828	0,791	0,695
Random Forest	HOG	**0**	0,878	0,999	0,935	0,963	0,871
		1	0,989	0,350	0,517	0,717	0,871
		Total	0,898	0,885	0,862	0,920	0,871
	BOVW-500	**0**	0,879	0,999	0,935	0,962	0,869
		1	0,982	0,355	0,522	0,713	0,869
		Total	0,897	0,886	0,863	0,918	0,869
	BOVW-1000	**0**	0,878	0,999	0,935	0,961	0,866
		1	0,983	0,349	0,515	0,712	0,866
		Total	0,897	0,885	0,861	0,917	0,866
	BOVW-2000	**0**	0,879	0,999	0,935	0,961	0,868
		1	0,984	0,354	0,521	0,714	0,868
		Total	0,898	0,886	0,862	0,918	0,868

Table A.23. 5-fold cross validation results on Word2Vec dataset with PCA with new split

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,911	0,596	0,720	0,918	0,744
		1	0,298	0,747	0,426	0,425	0,744
		Total	0,796	0,624	0,665	0,826	0,744
	BOVW-500	**0**	0,911	0,596	0,720	0,918	0,743
		1	0,298	0,746	0,426	0,424	0,743
		Total	0,796	0,624	0,665	0,826	0,743
	BOVW-1000	**0**	0,911	0,596	0,720	0,918	0,743
		1	0,298	0,746	0,426	0,425	0,743
		Total	0,796	0,624	0,665	0,826	0,743
	BOVW-2000	**0**	0,911	0,595	0,720	0,918	0,743
		1	0,298	0,747	0,426	0,425	0,743
		Total	0,796	0,624	0,665	0,825	0,743
k-NN	HOG	**0**	0,962	0,829	0,891	0,941	0,843
		1	0,537	0,857	0,660	0,491	0,843
		Total	0,882	0,835	0,847	0,857	0,843
	BOVW-500	**0**	0,962	0,828	0,890	0,940	0,842
		1	0,535	0,857	0,658	0,486	0,842
		Total	0,882	0,834	0,847	0,855	0,842
	BOVW-1000	**0**	0,962	0,828	0,890	0,940	0,842
		1	0,535	0,857	0,658	0,486	0,842
		Total	0,882	0,834	0,847	0,855	0,842
	BOVW-2000	**0**	0,962	0,828	0,890	0,940	0,842
		1	0,535	0,857	0,659	0,486	0,842
		Total	0,882	0,834	0,847	0,855	0,842
Random Forest	HOG	**0**	0,872	0,997	0,930	0,974	0,911
		1	0,968	0,364	0,529	0,779	0,911
		Total	0,890	0,879	0,855	0,937	0,911
	BOVW-500	**0**	0,871	0,997	0,930	0,974	0,909
		1	0,968	0,357	0,522	0,774	0,909
		Total	0,889	0,877	0,853	0,936	0,909
	BOVW-1000	**0**	0,871	0,997	0,930	0,974	0,909
		1	0,965	0,359	0,523	0,774	0,909
		Total	0,889	0,877	0,854	0,936	0,909
	BOVW-2000	**0**	0,871	0,997	0,930	0,973	0,907
		1	0,964	0,358	0,522	0,772	0,907
		Total	0,888	0,877	0,853	0,935	0,907

Table A.24. Test validation results on Word2Vec dataset with PCA with new split

			Results				
Classification	**Method**	**Class**	***Precision***	***Recall***	***F-measure***	***PRC-Area***	***ROC-Area***
Naive Bayes	HOG	**0**	0,909	0,596	0,720	0,914	0,736
		1	0,299	0,742	0,426	0,421	0,736
		Total	0,794	0,624	0,665	0,821	0,736
	BOVW-500	**0**	0,908	0,596	0,719	0,914	0,736
		1	0,298	0,740	0,425	0,420	0,736
		Total	0,793	0,623	0,664	0,821	0,736
	BOVW-1000	**0**	0,908	0,596	0,719	0,914	0,736
		1	0,298	0,741	0,425	0,420	0,736
		Total	0,794	0,623	0,664	0,821	0,736
	BOVW-2000	**0**	0,907	0,597	0,720	0,914	0,734
		1	0,298	0,737	0,424	0,418	0,735
		Total	0,793	0,623	0,665	0,820	0,734
k-NN	HOG	**0**	0,892	0,840	0,865	0,876	0,687
		1	0,447	0,559	0,497	0,325	0,687
		Total	0,808	0,787	0,796	0,773	0,687
	BOVW-500	**0**	0,892	0,862	0,877	0,864	0,655
		1	0,480	0,548	0,512	0,335	0,655
		Total	0,814	0,803	0,808	0,765	0,655
	BOVW-1000	**0**	0,893	0,870	0,882	0,862	0,650
		1	0,495	0,551	0,522	0,340	0,650
		Total	0,818	0,810	0,814	0,764	0,650
	BOVW-2000	**0**	0,891	0,870	0,880	0,864	0,654
		1	0,491	0,542	0,515	0,340	0,654
		Total	0,816	0,808	0,812	0,765	0,654
Random Forest	HOG	**0**	0,870	0,998	0,929	0,965	0,884
		1	0,973	0,355	0,520	0,732	0,884
		Total	0,889	0,877	0,852	0,921	0,884
	BOVW-500	**0**	0,870	0,998	0,930	0,965	0,883
		1	0,976	0,357	0,522	0,731	0,883
		Total	0,890	0,877	0,853	0,921	0,883
	BOVW-1000	**0**	0,870	0,998	0,929	0,965	0,883
		1	0,977	0,354	0,520	0,732	0,883
		Total	0,890	0,877	0,852	0,921	0,883
	BOVW-2000	**0**	0,870	0,998	0,930	0,964	0,881
		1	0,974	0,357	0,523	0,729	0,881
		Total	0,890	0,877	0,853	0,920	0,881