QUOTE EXAMINER: VERIFYING QUOTED IMAGES USING WEB-BASED TEXT SIMILARITY

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science and Engineering

Submitted By
Sneha Banerjee
(Roll No. 801732050)

Under the supervision of:
Dr. Parteek Kumar
Associate Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
PATIALA – 147004

June 2019

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "***Quote Examiner: Verifying Quoted Images Using Web-based Text Similarity***", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Parteek Kumar* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(Sneha Banerjee)

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

Parteek Kumar

(Dr. Parteek Kumar)
Associate Professor, CSED

ACKNOWLEDGEMENT

First of all, I would like to thank the Almighty, who has always guided me to work on the right path of the life. This work would not have been possible without the encouragement and able guidance of my supervisor **Dr. Parteek Bhatia**. I thank my supervisor for his time, patience, discussions and valuable comments. His enthusiasm and optimism made this experience both rewarding and enjoyable. I am also thankful to **Ms. Sawinder Kaur**, PhD scholar in Computer Science and Engineering Department for her guidance and valuable time.

I would like to express my gratitude to **Dr. Maninder Singh**, Head of Computer Science and Engineering Department and **Dr. Ashutosh Mishra**, P.G. coordinator for their constant motivation and encouragement.

I am also thankful to the entire faculty and staff members of Computer Science Department for their direct-indirect help, cooperation, love and affection.

Last but not the least, I would like to thank my parents for their wonderful love and encouragement, without their blessings none of this would have been possible. I would also like to thank my close friends for their constant support.

Sneha Banerjee

ABSTRACT

In recent times, there has been a rapid advancement in digital data mainly in visual formats, such as images from the web, mobiles, digital cameras, screenshots, *etc.* Images with quotes are spreading virally through online platforms like the internet, Facebook, WhatsApp, *etc*. Misquotations often spread like a forest fire through social media, which highlights the lack of responsibility of the web users when circulating poorly cited quotes. Thus, it is important to authenticate the text contained in the images being circulated online. Hence, there is a need to retrieve the information within such textual images. Optical Character Recognition (OCR) is a method used for converting textual images into readable text format. There are various OCR tools available which help in converting visual data into editable textual documents. In this study, a performance analysis between various OCR tools like Tesseract-OCR, Google Cloud Vision and AWS rekognition is presented on natural scene images. Further, a post-processing technique has been applied on the obtained text and it has been observed that after removing spelling errors from the identified text in images resulted in a significant improvement in the accuracy of the output text. There has been an improvement of around 2% in the case of natural scene images and approximately 8% in the case of text obtained from handwritten images. Additionally, it has been observed that in case of natural scene images, Google Cloud Vision gives an overall F1-score of 88.32%, AWS rekognition gives an overall F1-score of 68.1% and Tesseract-OCR gives an F1-score of 54.58%. Accordingly, it can be deduced from the results that Google Cloud Vision outperforms the other two tools in consideration and has therefore been used for extracting text from quoted images. In this experiment, a web-based text similarity approach has been used to examine the authenticity of the content of the quoted images. Google Custom Search Engine has been used to retrieve the URLs of the similar text followed by verification of the obtained domain names against authentic quotation sites. Approximately, 96.26% accuracy has been achieved in classifying quoted images as verified or misquoted by using the verification results.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

OCR	Optical Character Recognition
NLP	Natural Language Processing
NER	Named Entity Recognition
AWS	Amazon Web Services
URL	Uniform Resource Locator
WAMBy	Web-based Answer Mining Bot
TF-IDF	Term Frequency – Inverse Document Frequency
POS	Parts Of Speech
SVM	Support Vector Machines
CNN	Convolutional Neural Network

CHAPTER 1
INTRODUCTION

1.1 Introduction to Quoted Images

In today's world, social media is flooded with data and information and most of these data are in the form of scanned images, mobile images and images in any visual format like *.jpg, .jpeg, .png, .bmp, etc*. To process the information contained in visual format, there is a need to detect the textual content of the images. OCR technology is used to detect text from images automatically. Though OCR tools perform efficiently in text recognition of scanned documents, still the text detection and its recognition in natural scene images is a challenging issue. Quoted images are a perfect example of natural scene images. Identifying text from natural scene image is a major challenge in the field of visualization and pattern recognition due to the presence of noise, variation in background, lighting and font styles. Figure 1.1 is an example of quoted image and the text which will be retrieved from this image using the OCR tool will be as below.

"One best book is equal to hundred good friends but one good friend is equal to a library." -Dr. A.P.J. Abdul Kalam

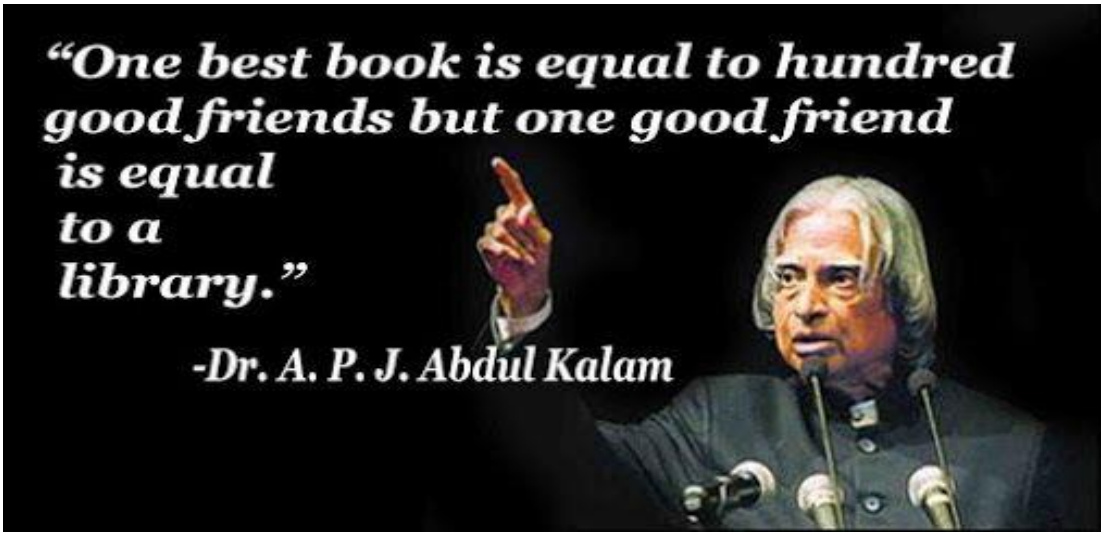


Figure 1.1: Example of Quoted Image [29]

Today's generation has shifted from books to online platform for gathering knowledge. However, not all online sources provide authentic information. In daily life, one tends to use quote found online without knowing whether they are valid or

not. Misquotes often spread like forest fire through social media, which highlights the lack of responsibility of the web users when circulating poorly cited quotes.

When one refers to famous quotes, there are various errors that are present all over the internet. Most of the quotes are divided into three categories as discussed below.

i. ***False Quotes*** – The famous quotes that were actually not said by the person being quoted, *i.e.*, the original source is not known. Figure 1.2 is an example of a false quote attributed to Albert Einstein, *"Insanity is doing the same thing over and over again and expecting different results"*. This quote has not been said by Albert Einstein and its original source is not known.

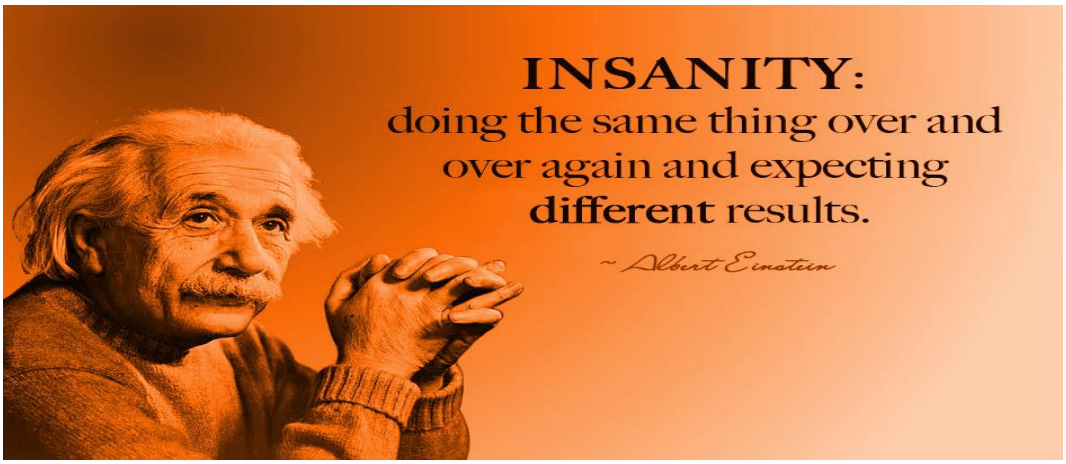


Figure 1.2: Example of False Quoted Image [29]

ii. **Misquotes** – The famous quotes that are not accurately presented, *i.e.*, there are errors in the original quote or the quote is rephrased. Figure 1.3 is an example of misquote, *i.e.*, *"Dreams are the royal road to the unconscious"* by Sigmund Freud. The actual quote said by him was *"The interpretation of dreams is the royal road to a knowledge of the unconscious activities of the mind"*.

Figure 1.3: Example of Misquoted Image [29]

iii. **Mis-attributed Quotes** – The famous quotes that have been attributed to the wrong person, *i.e.*, a more famous person gets the credit of the quote. Figure 1.4 is an example of a mis-attributed quote, *"In the end, it's not the years in your life that count. It's the life in your years"*, which was actually not said by Abraham Lincoln but by Edward J. Stieglitz.

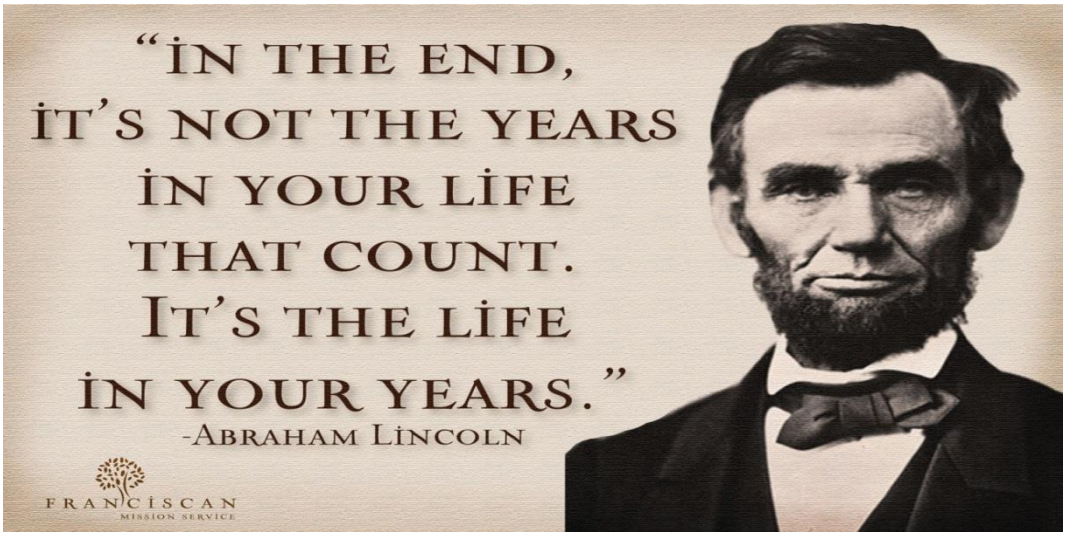


Figure 1.4: Example of Mis-attributed Quoted Image [29]

Quotes usually find their way into Facebook profiles, WhatsApp and Twitter posts, where they get multiplied and often get distorted in the retelling, ultimately leaving the end-readers with misquoted materials (normally void of any context). Hence, there is a need to verify quotes before its usage in order to differentiate a fake or misquote from an authentic one.

The flow of the proposed approach has been highlighted in the Figure 1.5. The flow involves reading of the quoted images using OCR tools which are discussed Section 1.2 and 1.3, followed by post-processing the retrieved text to remove the grammatical errors. Further, the retrieved text is provided as input to two phases, *i.e.*, for author name recognition and other for verification and classification of the quote into categories *Verified* or *Misquote.* The process of classification and verification involves using text similarity approach as explained in Section 1.4 and the process of NER has been described in Section 1.5.

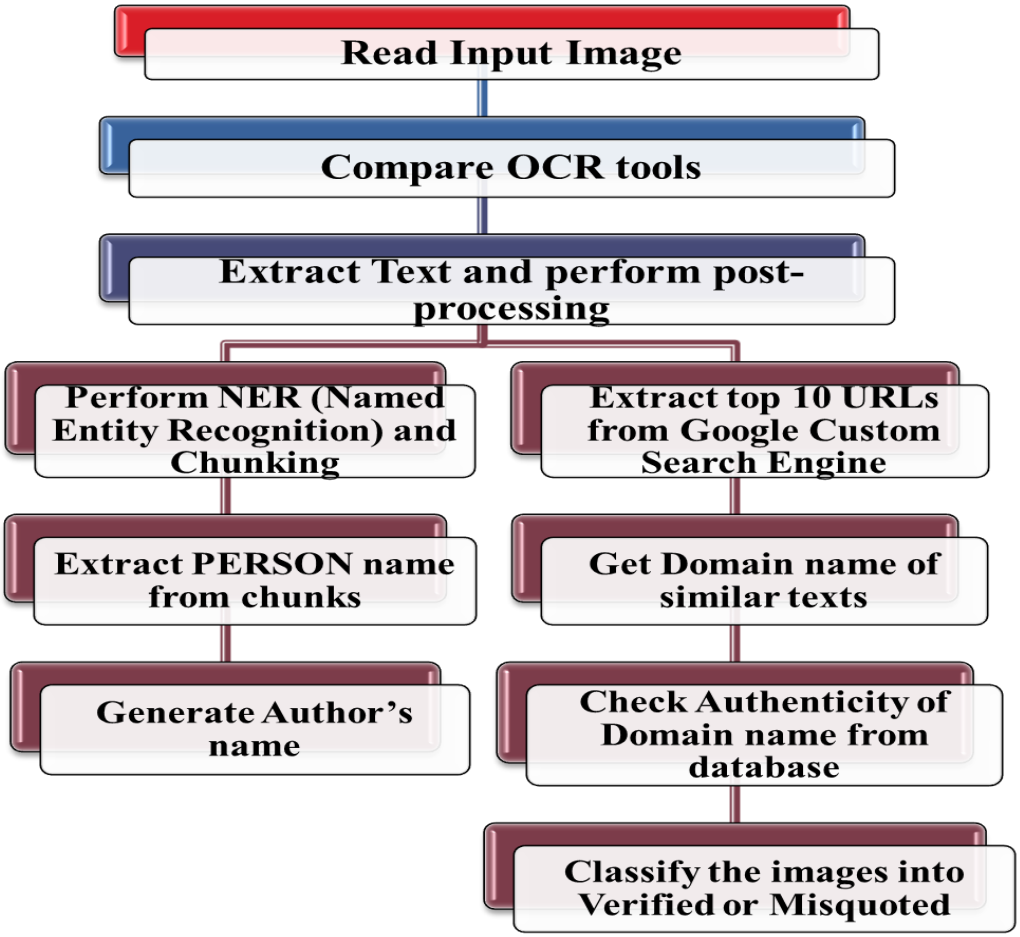


Figure 1.5: Flow Diagram of the Proposed Approach

This research focuses on text detection from quoted images and verification of its content. For this purpose, quotes are identified from images using Google Cloud Vision, an online OCR tool. Additionally, a post-processing technique has been applied to the extracted text to automate the removal of spelling errors, thus leading to improved accuracy in the task of text retrieval from images. Further, there is a need to verify whether the origin of the information obtained is valid or not in order to classify the image as containing verified quote or not. A web-based text similarity search is performed using Custom Google Search Engine to retrieve the source (URLs and domain name) of the text retrieved in order to check the validity of the source. Lastly, a novel classification approach has been performed to classify the extracted text into categories, *Verified* or *Misquoted*. Further, NLP technique has been used to recognize the author's name from the quoted image. In this research, images with quotes by famous celebrities and global leaders have been used; therefore there is a need to retrieve the author's name along with the quotation from the image. For this purpose, NLP technique called Chunking and NER has been used.

1.2 Optical Character Recognition

Optical Character Recognition (OCR) is a method used for converting textual images into readable text format. There are various OCR tools available which help in converting visual data into editable textual documents. With the help of OCR, scanned documents are no more just image files, instead become entirely editable documents which can be stored in computers. With use of OCR, people need not manually enter and type information into documents, rather OCR recognizes appropriate data and enters it automatically. Figure 1.6 highlights the flow of OCR process in recognizing information from a scanned document to auto-populate the required document. The working of OCR tool from Figure 1.6 can be summarized as below.

- A scanned document or any image serves as an input to the OCR tool.
- The text contained in the input image is recognized by the OCR tool. In the case of Figure 1.5, text contained in the scanned document is identified. The recognized text includes information like, name of the person, his/her date of birth, contact details, *etc.*
- The extracted text is then stored in the required format in computer which is easily accessible to the end-user.
- This method reduces human effort of manual filling of online forms and automates the procedure.

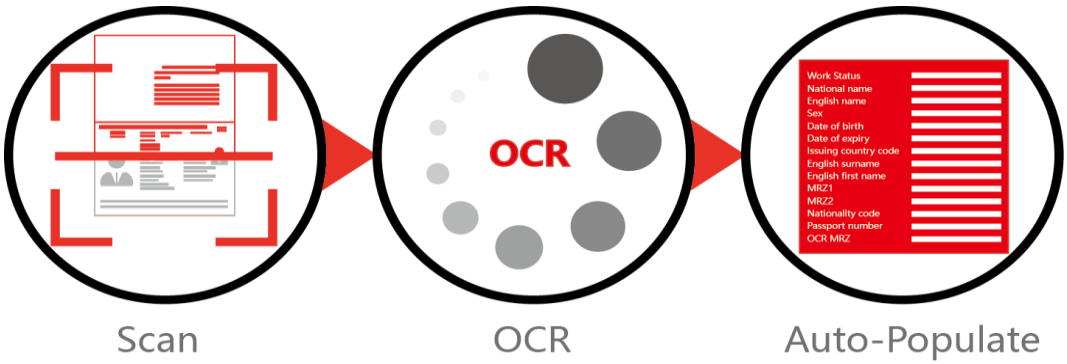


Figure 1.6: Flow of OCR process [29]

Apart from detecting text from images being circulated online, OCR can be utilized in various other fields which have been discussed below.

- ***Banking:*** In the field of banking, OCR can be used for analyzing handwritten cheques and recognizing and verifying signatures. This reduces the overall time for cheque clearance.
- ***Legal:*** All the legal documents, affidavits, wills, *etc.* can be digitized and stored for further use with the help of OCR tools.
- ***Healthcare:*** Using OCR, one can create an entire medical history as a digital storage having information such as patient's history and treatments, diagnostic tests, medical archives, *etc.* This helps in maintaining the medical records as a unified storage, instead of managing heaps of files and reports.
- ***Education:*** The most popular benefit of using OCR in the field of education is the availability of e-books and various texts that are available online.
- ***Other Industries:*** To keep track of financial records and to avoid piling up of payments, various industries use invoice imaging application. This invoice imaging requires OCR for information retrieval. Similarly, other applications of OCR include handwritten recognition, barcode detection, simplification of data collection and analysis.

1.3 OCR Tools

OCR involves recognition of text from scanned images, mobile images, and from any visual image format. One such OCR tool available as an open-source software is Tesseract-OCR. Tesseract was released under the Apache License, and its development was sponsored by Google since 2006. Initially, Tesseract was the most accurate open-source OCR engine available. This tool is accurate in detecting text from scanned images, however, its performance degrades in case of complex images like images having different styles and fonts, colored images, quoted images, *etc.* Therefore, there is a need to use better tools for text detection in images. Along with Tesseract-OCR, in this thesis, Google Cloud Vision and Amazon Web Services (AWS) rekognition are used for text recognition from images. Both these tools are online OCR tools.

A detailed analysis of the performance of all these three tools- Tesseract-OCR, Google Cloud Vision and AWS rekognition on natural scene images has been documented in Chapter 6. Unfortunately, all the tools mentioned sometimes result in

the false recognition of the characters in the images leading to spelling and linguistic errors. Therefore, a post-processing technique has been applied on the extracted text to automate the removal of spelling errors, thus leading to improved accuracy in the task of text retrieval from images. It has been observed that Google Cloud Vision performed better in identifying text from natural scene images and hence has been used for identifying text from the quoted images.

1.4 Web-based Text Similarity

Text similarity refers to how similar two chunks of text are both in terms of surface/lexical/word level similarity and meaning/semantic similarity. For example, how much alike are the phrase *"the dog ate the rat"* and *"the rat ate the dog food"*. Considering only lexical similarity, these two phrases are quite similar as out of total 4 unique terms, 3 are exactly same. However, it is known that though the words significantly overlap, still the meaning of each phrase is entirely different. This can be deduced using semantic similarity approach. Semantic similarity usually deals with Natural Language Processing (NLP) tasks like *paraphrase identification* and *automatic question answering*.

The most popular methods to perform text similarity are as follows.

i. Jaccard Similarity: It is defined as intersection over union and is calculated using the formula given in 1.1.

$$\text{Jaccard similarity} = \frac{\text{Size of intersection of two sets}}{\text{Size of union of two sets}} \tag{1.1}$$

For evaluating Jaccard similarity, each word is converted to its root word using lemmatization. For example, root word for both *"happiness"* and *"happily"* will become *"happy".* Also, in this case if two different words get reduced to the same root word, then it will be counted as one unique element of the set.

ii. Cosine Similarity: It measures similarity by calculating cosine of angle between two vectors. Thus, there is a need to convert the sentences into vector format. For this purpose, two approaches are generally used which are discussed given in 1.2.

a. Bag of Word using TF-IDF: TF-IDF or term frequency- inverse document frequency is used to convert and store each word as a number. It is useful for classification of the documents as a whole.

b. Word2Vec or Word embedding: In this one vector for each word is created and it is good for identifying contextual content.

The cosine similarity for two texts *x* and *y* can be calculated using the below formula, where *X* and *Y* represent the vector form of the texts *x* and *y* respectively, *n* represents the total number of unique words. X_i represents the frequency of word *i* in the text *x*.

$$\text{Similarity or } \cos(\theta) = \frac{X.Y}{|X||Y|} = \frac{\sum_{i=1}^{n} X_i.Y_i}{\sqrt{\sum_{i=1}^{n} X_i}.\sqrt{\sum_{i=1}^{n} Y_i}} \quad (1.2)$$

Taking example of two quotes, one verified and the other misquote, let's see what information can be retrieved using the above two methods.

Quote 1: *Dreams are the royal road to the unconscious - Sigmund Freud*

Quote 2: *The interpretation of dreams is the royal road to a knowledge of the unconscious activities of the mind - Sigmund Freud*

Total number of unique words is 15 and the two quotes have 8 words in common. Evaluating similarity metrics, below results are obtained.

Jaccard similarity = 8 / 15 = 0.533

Cosine similarity = 16 / $\sqrt{12}$ * $\sqrt{38}$ = 0.749

From the obtained results, only the extent of similarity between the two quotes can be deduced but no information about its authenticity or source can be retrieved. Moreover, as the text is retrieved using OCR a small spelling error can change the value of similarity to a great extent. Therefore, there is a need to use a web-based approach which provides information about the source of the text. The sub-section 1.4.1 discusses about the Custom Google Search engine and its importance in extracting information about the originating source of any textual content.

1.4.1 Custom Google Search Engine

It is a Google platform which allows web users to incorporate specific information in web explorations, improve and classify queries and design customized search engine, same as Google Search. Figure 1.7 shows the home page of the Custom Google Search Engine.

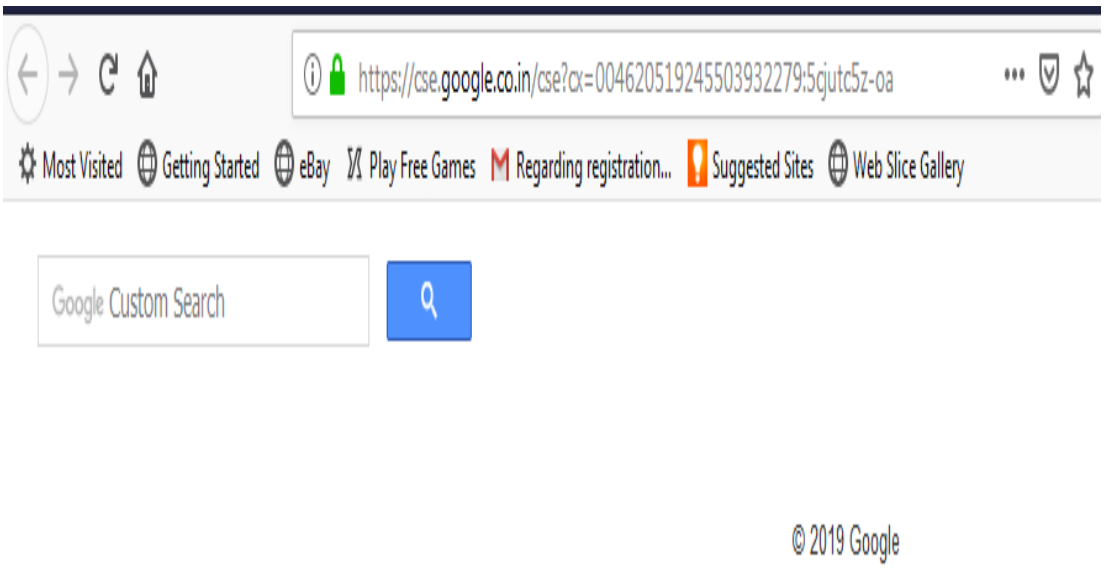
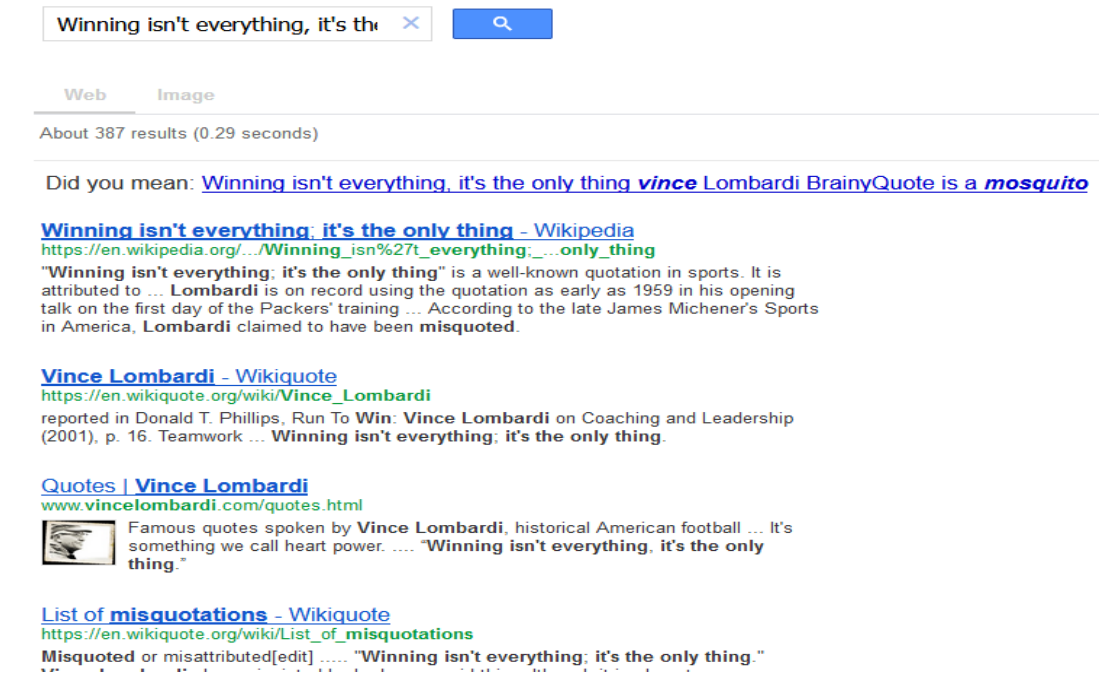


Figure 1.7: Custom Google Search engine home page [29]

If one provides an authentic and misquote as search query then the retrieved results, *i.e.*, URL and domain names shall provide some relevant information about the source of the quotation. Figure 1.8 (b) shows the information retrieved for text extracted from the input provided in Figure 1.8 (a).

(a) Example of a misquote



(b) Custom Google Search Engine result

Figure 1.8: Working of Custom Google Search Engine [29]

It is visible from the obtained result that the provided input is among the list of misquotation. However, there are systems like *Google Reverse Image Search Engine* to retrieve source of an image but it cannot retrieve source of the textual content present in any image. Figure 1.9 shows result obtained when the Figure 1.8 (a) is provided as input to Google Reverse Image Search engine.

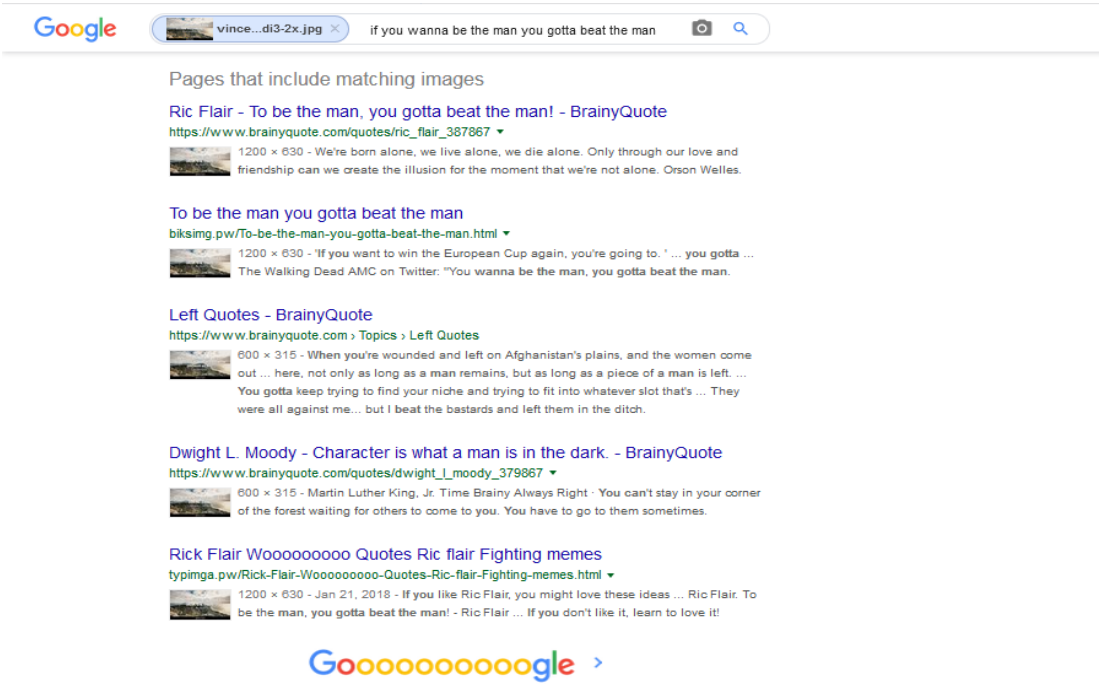


Figure 1.9: Working of Google Reverse Image Search Engine [29]

It can be observed from the obtained result that Google Reverse Image Search is helpful in detecting origin of any image but cannot be relied for retrieving the source of the text contained in any image. Thus, for performing web-based text similarity Google Custom Search is the preferred choice.

1.5 Named Entity Recognition

The main focus in this research is on quoted images. For any quotation, the author name is an important feature. In order to identify author's name present in the quoted image, NLP techniques have been used specifically Named Entity Recognition (NER), also known as entity extraction. It is a traditional procedure used in information retrieval to recognize and segment the named entities and classify them into several predefined categories. All the NLP techniques involved for applying NER have been discussed below.

- ***Tokenization:*** There are two types of tokenizations available, namely word tokenization and sentence tokenization. In the first process a text is broken down into words, known as tokens by identifying spaces between the two words. The latter process includes breaking down of text into individual sentences by identifying sentence terminator, *i.e.,* the symbol *"."*. Therefore, for identifying author's name, sentence tokenization is performed followed by word tokenization.

- ***POS tagging:*** The part of speech (POS) describes how a term is used in a sentence. There are mainly eight POS tags available - noun, pronoun, adjective, verb, adverb, preposition, conjunction and interjection. After tokenization, each token is provided with one of the POS tags for further use. For example, *"Albert"* is provided with tag NNP which represents the proper noun.

- ***Chunking:*** Chunking is a method of selecting phrases from the unorganized text. Instead of using a single token with just tags that do not describe the exact significance of the text, it's desirable to use phrases such as *"Albert Einstein"* as a single term rather than using *'Albert'* and *'Einstein'* as individual terms.

  Chunking operates on top of POS tagging. The process utilizes POS-tags as information and produces chunks as output. Just like POS tags, a standard set of Chunk tags like *Noun Phrase (NP), Verb Phrase (VP), etc.* is present. It is an important step when information from text such as Name of person, location, organization, *etc*. is required.

For example, "*Albert Einstein*" is identified as Noun Phrase using chunking. On applying NER on the same, it will be tagged PERSON indicating it is a person's name.

1.6 Text Classification

Text classification is the method of allocating labels to text depending on the content of the string. It is one of the primary tasks in NLP having variety of applications like opinion mining, subject labeling, spam and intent detection, *etc.* Unstructured data especially textual data is present throughout the social media and the internet. Such text can be a remarkably precious source of information; however, deriving knowledge from it is usually difficult and exhausting.

Text classifiers are applied to design, build, and categorize any textual content. A classifier takes text as an input, analyzes its essence, and then accordingly assigns a relevant tag to it. Text categorization can be performed in two ways, *i.e.*, manually and automatically. In the manual procedure, a human annotator understands the meaning of the text and classifies it accordingly. This process provides quality outcomes but is tedious and expensive. The latter uses machine learning, NLP, and other methods to automatically analyze and categorize the text in a faster and profitable way.

Several machine learning models, such as Logistic Regression, Naïve Bayes, Support Vector Machines (SVM), *etc.* are popular for text classification. To categorize text containing authentic or fake content, machine learning and deep learning models are widely used approach. For this purpose, a proper training model needs to be prepared using a training dataset containing required features. In the proposed approach, the training dataset contains quotes labelled as being Verified or Misquote. Using the training dataset, the machine learning models are trained for classifying quotes into *Verified* or *Misquote*.

All the traditional machine learning methods used in this research have been discussed briefly below.

- ***Logistic Regression:*** Logistic regression is a suitable regression analysis to manage when the target variable is dichotomic, *i.e.*, binary. Similar to other regression analysis, the logistic regression is a predictive analysis. It is used to

represent data and describe the association within one dependent or target binary variable and one or more independent variables.

- ***Naïve Bayes:*** It is a statistical algorithm that is used for text classification. Multinomial Naïve Bayes (MNB) is a member of the same family and its advantage is that it generates remarkably great results for small data (a couple of 1000tagged examples) and computational resources are limited. It is based on Baye's Theorem which functions on conditional probabilities.
- ***Support Vector Machines (SVM):*** Similar to Naïve Bayes, SVM does not require large training data for text classification. However, it requires more computational resources and can generate more accurate results compared to Naïve Bayes. From literature, it has been observed that SVM classifiers are preferred for pattern recognition and classification when there are exactly two classes. The objective of the SVM algorithm is to find a hyperplane in an N-dimensional space where N represents the number of features that distinctly classifies the data points.

In this research, a text classification technique has been proposed which uses web-based text similarity approach and NLP to classify the quoted images into categories, *Verified* and *Misquoted.* This approach has been used to improve the performance of the classification process.

1.7 Thesis Outline

The main aim of this thesis is to recognize text from quoted images and verify the authenticity of its content. In this chapter an overview of the research has been documented. Chapter 2 highlights the various works done in the field of text detection from images, text similarity and identification of fake contents. Chapter 3 discusses the problem statement. In Chapter 4, the working of the proposed system has been explained. Further, Chapter 5 illustrates the design and implementation of the proposed system. The obtained results have been highlighted in Chapter 6 followed by conclusion and future scope in Chapter 7.

CHAPTER 2
LITERATURE REVIEW

In this chapter, an overview of the approaches related to the research areas included in the thesis work has been presented. The main research areas in the thesis are text detection from images, web-based text similarity and identification of fake contents which are discussed in the sub-sections 2.1, 2.2 and 2.3 respectively.

2.1 Text Detection from images

Text identification from images is an ongoing research field of pattern recognition. To analyze the issues related to text retrieval of images, various researchers have suggested several technologies, every approach or method attempts to focus on the problems involved differently. Figure 2.1 gives an overview of the traditional OCR system components. The traditional OCR performs the below steps.

- Scan the input image.
- Performs binarization, *i.e.*, pre-process the image by converting it into black and white.
- Segmentation is performed which includes identifying the fragment of images containing textual data.
- Feature extraction refers to the process of identifying each character using the respective feature set or characteristic of the character.
- Recognition refers to the ultimate text detection by aggregating the characters recognized from the entire image.
- Final step in OCR system is to store the recognized text into computer readable format.

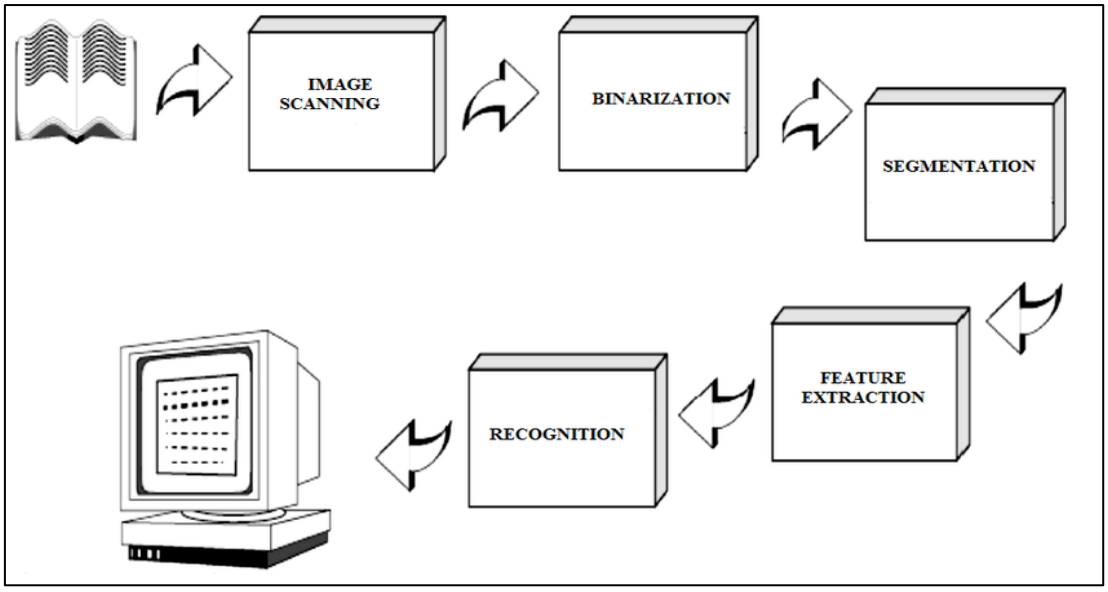


Figure 2.1: A traditional OCR system components [2]

2.1.1 Related Work on Text Detection from Images

Gur *et al.* (2012) discuss automated optical character recognition (OCR) tools. It shows that OCR tools do not provide a complete solution and in many cases, human inspection is required for accuracy achievement. The authors have suggested a novel text recognition algorithm based on fuzzy logic rules [1] and its usage, relying on statistical data of the analyzed font. This approach enables the recognition of distorted letters that may not be retrieved otherwise, by combining letter statistics and correlation coefficients in a set of fuzzy based rules. The authors focused on calligraphy of the handwritten Rashi fonts associated with commentaries of the Bible.

Dutta *et al.* (2012) have proposed a novel research methodology resulting in a 15% decrease in word error rate on highly degraded document images [2] written in Indian languages. AlSalman *et al.* (2012) propose a Braille recognition technique [3], which helps to detect and identify Braille characters embossed on Braille documents. The result is utilized in various applications like embossing, printing, translating, *etc.* Due to poor quality imaging, the performance of these applications is altered and damaged. The reduced quality imaging is due to many factors like scanner quality, scan resolution, lighting, and various types of embossed documents.

Du *et al.* (2012) proposed Automatic license plate recognition (ALPR) system [4] where the information about the vehicle license plate is extracted from a single image or a series of images. The ALPR uses a color, black and white, or an infrared camera to take images. Figure 2.2 shows the working of ALPR system.

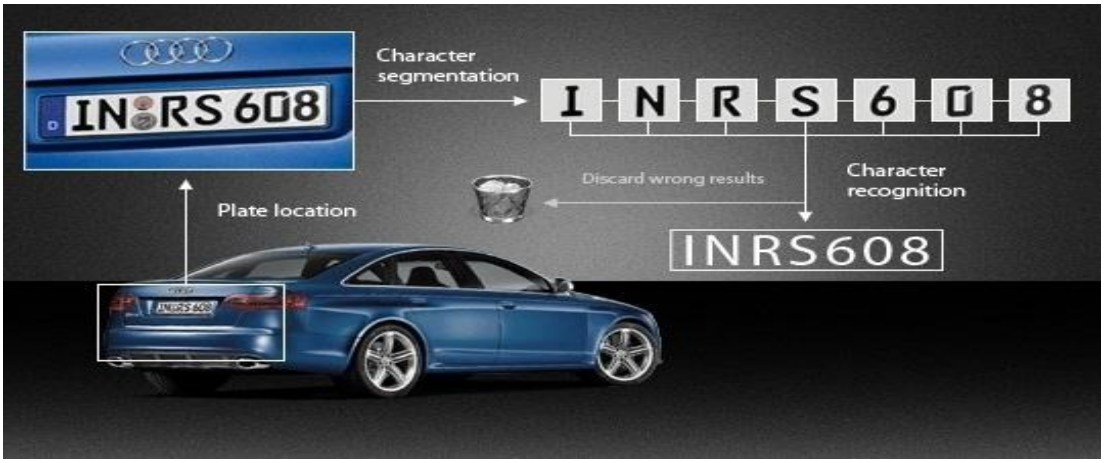


Figure 2.2: Working of ALPR system [4]

An innovative adaptive binarization approach based on wavelet filter [5] has been proposed by Yang *et al.* (2012), showing comparable performance to other similar methods. However, the proposed approach processes faster making it more suitable for real-time processing and easily applicable to mobile devices. This approach makes use of complex scene images of ICDAR 2005 Robust Reading Competition for evaluation.

Bassil *et al.* (2012) proposes a post-processing algorithm for error correction in text detected using OCR tools by using Google's online spelling suggestion [6]. This method helps in detecting and correcting the OCR real-word and non-word errors. The proposed approach reveals a significant improvement in the OCR error correction rate.

Nitrogenous *et al.* (2013) highlights the importance of image binarization phase [7] in document image analysis and identification. The binarization of an image affects the later stages of document image analysis and identification pipeline. The proposed approach helps in evaluating the images of historical printed/handwritten documents by using pixel-based binarization method.

Manwatkar *et al.* (2015) propose a text recognition method to perform Document Image Analysis (DIA) [8] which is a technique of transforming documents in paper format into an electronically readable format. This method involves scanning of printed documents, converting them to grayscale and binary format, followed by line and then character detection to finally obtain the output text.

Rajan *et al.* (2017) propose a hybrid text detection method from natural scene images. The proposed method uses a fractional Poisson model [9] to improve the quality of

the images obtained by Laplacian operation. The enhanced image looks brighter than input and Laplacian images and the image does not contain any noise.

Kushol *et al.* (2018) has proposed a new technique to retrieve text from mobile images using Google Cloud Vision [10]. This experiment gives efficient and effective results in terms of accuracy as well as processing time.

In this thesis, a comparison of OCR tools, namely Tesseract-OCR, Google Cloud Vision and AWS rekognition have been performed. Figure 2.3 presents the flow of the Tesseract-OCR tool which consists of using scanned images as input followed by pre-processing. The pre-processing phase includes reducing noise from image along with its sharpening and smoothing. This phase is followed by recognizing the individual characters from the image and classifying it to its respective character class. A post-processing step using Google's online spell check has been also performed. Lastly, the recognized text is stored in a computer editable format.

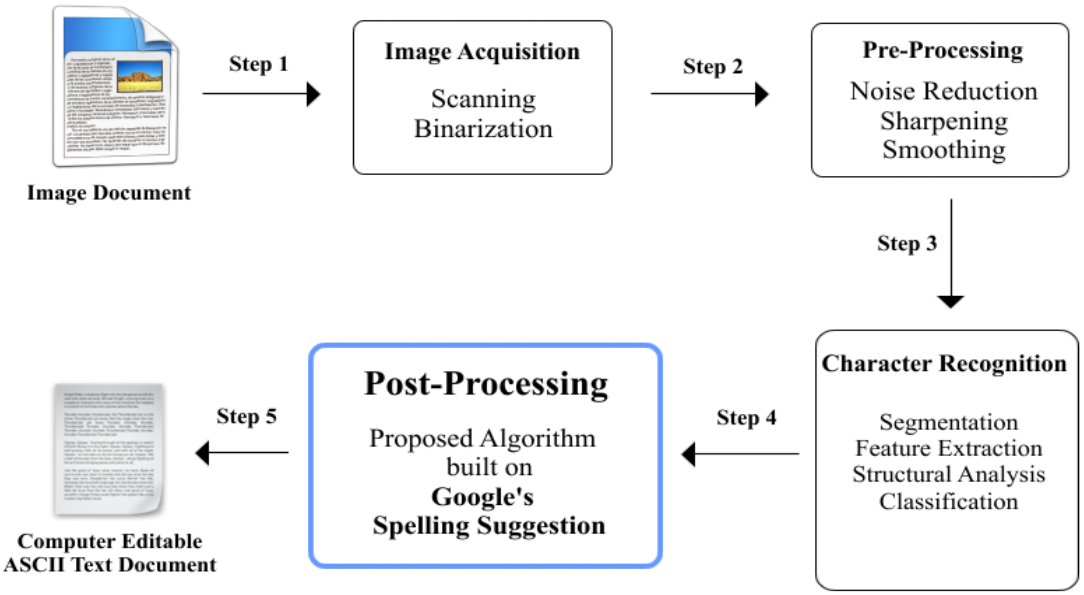


Figure 2.3: Flow of Tesseract-OCR enhanced with post-processing [6]

The entire flow of the system proposed by Kushol *et al.* using Google Cloud Vision has been depicted in Figure 2.4 which includes capturing the image using camera. Next step is pre-processing the image followed by performing OCR, *i.e.*, detecting the text from the image. Further, NLP is performed on the detected text for extracting important information from the image.

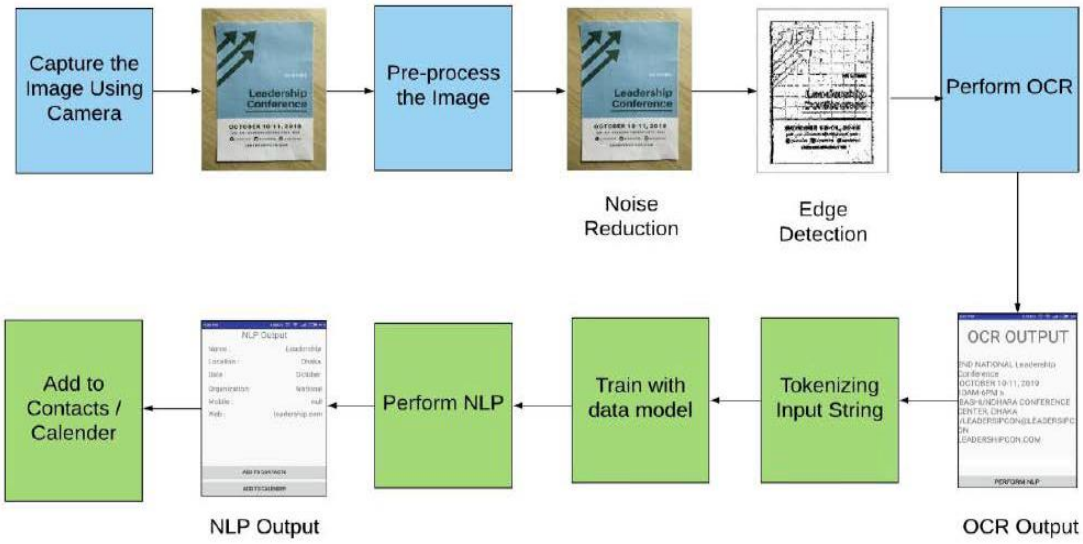


Figure 2.4: Flow of application using Google Cloud Vision [10]

AWS rekognition has been particularly used to deal with real-world images instead of document images. It is used to recognize and extract textual content from images which has been depicted in Figure 2.5. It encourages text in most Latin scripts and characters embedded in a wide category of designs, fonts, and styles. It even promotes identification of text superimposed on background articles at several orientations, like banners and billboards.

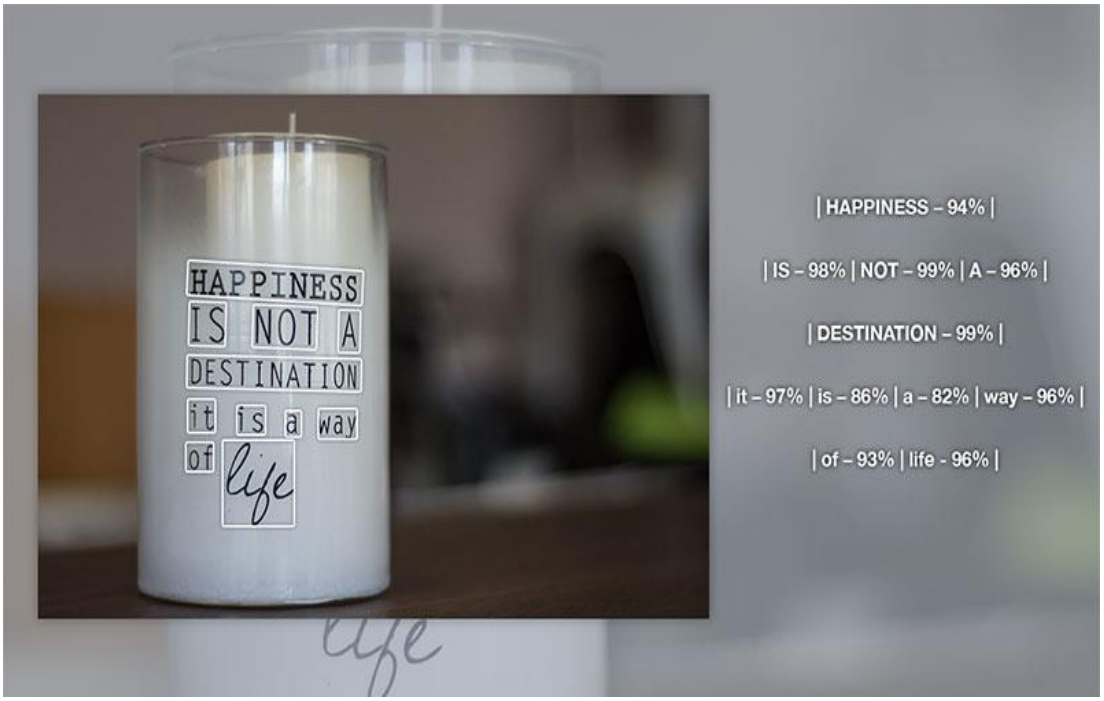


Figure 2.5: Text detection using AWS rekognition [30]

2.2 Text Similarity

With an enormous amount of textual data available on the web, there is an increasing need to be able to fetch the most suitable document. Estimating the similarity among terms, sentences, statements, and texts is an essential element in numerous tasks like information retrieval, document clustering, word-sense disambiguation, text summarization, *etc.* The procedure and various methods involved in the field of text similarity have been discussed in Chapter 1, section 1.4.

2.2.1 Related Work on Text Similarity

Gomaa *et al.* (2013) discusses the existing literature on text similarity [11] by partitioning them into 3 methods; String-based, Corpus-based and Knowledge-based similarities. Moreover, examples of a combination of the mentioned similarity methods are described.

Pradhan *et al.* (2015) highlights using various text similarity approaches like lexical, semantic and fusion similarity [12] for retrieving the most relevant document corresponding to a user's query. The fusion similarity is a hybrid approach with the combination of kernel based similarity and cosine based similarity.

Samarinas *et al.* (2018) proposed to generate a question answering system (WAMBy) which extracts answers from top 10 Google search results [13]. A novel text-similarity approach to improve performance of TF-IDF in synonyms and paraphrase identification has been suggested in [13].

Kowser *et al.* (2019) proposed developing a text similarity method which allows scholars to examine a text by analyzing with a regular response recorded in the system. The aim of the system is to reduce the time and also abolish the likelihood of biases using traditional similarity measures like TF-IDF, Word2Vector, Euclidean [14], Manhattan, Minkowski, Cosine and Jaccard Similarity.

2.3 Identification of Fake content

To identify fake contents like fake news, fake reviews, *etc.* classification of text is required. The various machine learning models are used for text classification. The models are discussed below.

i. Logistic Regression: Logistic regression designs the probability of the default class. For example, in case a person's gender is to be modelled as male or female

using their height, then the default class can be male and the logistic regression model can be formulated as the probability of male given a person's height, *i.e.*, *P (gender = male | height)* . Written in a different way, the probability that an input (A) belongs to the default class (B = 1) can be formulated using the formula given in 2.1.

$$P(A) = P(B=1|A) \tag{2.1}$$

Logistic regression is a linear system; however, the predictions are modified using the logistic function. The model can be stated as the formula given in 2.2.

$$P(A) = e^{(b0 + b1*A)} / (1 + e^{(b0 + b1*A)}) \tag{2.2}$$

ii. Naïve Bayes: Naïve Bayes classifiers are a combination of classification algorithm based on Bayes' Theorem. It is not an individual algorithm but a class of algorithms where all of them yield a general principle, *i.e.*, each pair of features being classified is independent of each other. Mathematically, Bayes' theorem can be evaluated using the formula given in 2.3.

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \tag{2.3}$$

Using naive assumption to the Bayes' theorem, *i.e.*, independence between the features, P(B) becomes 0. Thus, if any 2 events A and B are independent, then, Naïve Bayes theorem can be calculated using the formula given in 2.4.

$$P(A,B) = P(A).P(B) \tag{2.4}$$

iii. Support Vector Machine (SVM): SVM is a discriminative classifier represented by a separating hyperplane. Using a labeled training data, the algorithm results in an optimal hyperplane that classifies new samples. In 2-D space, this hyperplane is a line separating a plane into 2 sections wherein each class lay in either side. SVM contains tuning parameters as described below.

- ***Kernel:*** SVM uses a collection of mathematical functions defined as the kernel. The purpose of the kernel is taking data as input and transforming it into the expected form. Various SVM algorithms apply different kinds of kernel functions namely linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.

- ***Regularization:*** This parameter determines that optimization needs to be done in SVM to avoid misclassifying each training sample. For SVM classification, there is a need for risk minimization for the equation 2.5.

$$C \sum_{i=1,n} L(f(x_i), y_i) + \hbar(w) \quad (2.5)$$

  - C is used to set the amount of regularization.
  - L is a loss function of the samples and the model parameters.
  - $\hbar$ is a penalty function of the model parameters.
- ***Gamma:*** This parameter determines how far the impact of a single training sample reaches. With low gamma, points far apart from the expected separation line are considered in the estimation for the separation line. Whereas high gamma indicates the points near to a plausible line are considered in the computation.

2.3.1 Related Work on Text Classification

Mukherjee *et al.* (2013) proposed a novel approach to detect fake review on Yelp real-life data [15] using SVM 5-fold cross validation to improve the classification of real-life spam data. Iyyer *et al.* (2015) propose a deep neural network [16] that struggles with and sometimes outperforms various deep learning models for NLP on sentiment analysis and factoid question answering tasks taking only a fraction of the training time.

Tripathy *et al.* (2015) present a comparative study of Naïves Bayes (NB) and Support Vector Machine (SVM) classification [17] algorithm. These algorithms have been used to categorize a sentimental article having either a positive or a negative review. The dataset used for training and testing the model is marked based on polarity movie dataset and a comparison with results of the existing literature is made for significant research.

Arras *et al.* (2017) proposed a topic categorization model using SVM and CNN (Convolutional Neural Network) [18]. Although the SVM and CNN models function likewise considering the classification accuracy, the latter displays a higher level of explainability making it more understandable for individuals and possibly more beneficial for other purposes.

Rosas *et al.* (2017) focus on the automatic recognition of fake content in online news. Two unique datasets for the task of fake news detection have been created which covers 7 separate news fields. One dataset is achieved by crowdsourcing [19] and the second one using web covering celebrities. Classification models are developed by aggregating lexical, syntactic, and semantic information, and also features describing the text readability qualities.

2.4 Methods and Tools Used

Many researchers have proposed various tools and methodologies in the field of text detection from images. Also, OCR post-processing has been proposed by Bassil *et al.* (2012) using Google's online spell checker. Kushol *et al*. (2018) have proposed using Google Cloud Vision for text detection to improve over traditional methods. Tripathy *et al.* (2015) recommends using SVM and Naïves Bayes for text classification. On the other hand, Arras *et al.* (2017) and Rosas *et al.* (2017) suggests using CNN and deep neural network for classification of text. Samarinas *et al.* (2018) proposed using web-based text similarity approach which is an improvement over classical methods like correlation and TF-IDF. Table 2.1 summarizes various methodologies used by researchers in the field of text detection and text similarity.

Table 2.1. Summary of the related work done

Author	Proposed System	Dataset	Model
Dutta *et al.* (2012) [2]	Robust Recognition Of Degraded Documents Using Character N-Grams	Scanned books and newspapers for Malayalam (100K words) and English dataset (140K words)	Approximate Nearest Neighbor was used for classification of 33K words into 1000 classes.
Du *et al.* (2012) [4]	A survey on Automatic License Plate Recognition (ALPR) system	An annotated dataset of car images having license plates.	Survey on various classification, edge detection and feature extraction models.
Bassil *et al.* (2012) [6]	To automate the proofreading of OCR text and provide context-based detection and correction of OCR errors.	Two low quality image documents, one in English (for which Google.com was used as spellchecker) and the other in Arabic (using Google.ae).	The proposed algorithm was implemented using MS C# 4.0 under the MS.NET Framework 4.0 and MS Visual Studio 2010.

Author	Proposed System	Dataset	Model
Skalban. *et al* (2012) [18]	To generate a set of questions using NLP techniques which can be used as pre-questions to support the creators of assessment materials.	A Documentary video on "Nuclear Fusion" with subtitles. The dataset was in the form of screenshots of the videos with timestamp	Compare the result of the pre-questions generated using images with the pre-questions generated using text-based system and both the systems with pre-questions generated manually.
Kushol *et al.* (2018) [10]	Extracting textual information from mobile images, i.e. contact information from business cards as well as event information from magazines, posters or flyers.	200 mobile images scaled to 1024x768 pixels and a training dataset of 1500 sentences to make a model for NER (Named Entity Recognition)	Google Cloud Vision API is used to retrieve text from captured images Apache OpenNLP to extract useful information, using its toolkit called NER.
Samarinas *et a*l. (2018) [19]	To generate a question answering system (WAMBy) which extracts answers from top 10 Google search results.	TREC tracks dataset : labeled 5500 questions SQUAD dataset: an unlabeled reading comprehension dataset.	LSTM for question classification TF-IDF variant for textual similarity

In this chapter, various work done in the field of text recognition from images, text similarity and verification of fake content have been discussed. It has been observed that text recognition from images is an active research field of pattern recognition. Also, various text similarity methods have been discussed and it has been observed that web-based text similarity helps to overcome the drawbacks of traditional methods. Additionally, it has been observed from literature that several machine learning models have been used for identifying and classifying fake contents. In this research, a novel classification approach has been proposed using web-based text similarity.

CHAPTER 3
PROBLEM STATEMENT

The Internet is breeding ground for the generation of misquotes. Short quotes gain their access into Facebook profiles, WhatsApp and Twitter posts, where they increase across the Web unencumbered by citation and authentic context. With online distributing intricate and precise quotes get distorted in the retelling, leaving end-readers with misquoted matter void of any content. Surprisingly, the media is usually guilty similar to any average Web user. Due to various groups of quote cites that make a limited effort to check quotes before posting them, it is challenging to differentiate a fake quote from an authentic one. Moreover, no online platform is available which checks the authenticity of the textual content circulated in the form of images. Thus, this thesis proposes a novel approach which classifies a fake quote from an authentic one which is being circulated in image format through online platform.

This thesis focuses on retrieval of the text and author's name from the quoted images and verifies the source and authenticity of its content. For this purpose, a web-based system *"Quote Examiner"* has been proposed which helps in retrieving text from the quoted images and then an automated spell checking is performed on the retrieved text to improve the accuracy of the text retrieved. Next, a web-based text similarity search is performed using Custom Google Search Engine to retrieve the source of the text retrieved in order to check the validity of the source. Ultimately, using Natural Language Processing, author's name is obtained from the retrieved text.

3.1 Research Gap

The approaches used in [2] [3] [4] [5] suggests using the OCR technique for recognizing textual contents of the images. The proposed approaches in [7] [8] [9] focus on improving quality of images before performing text detection. The work proposed in [1] and [6] highlights the drawbacks of the OCR tools in accurately detecting the text from images. Using an online OCR tool, Google Cloud Vision API, for better text recognition has been suggested in [10]. Therefore, there is a need to improve the accuracy of text identification of images. Hence, in this research, the

performance of OCR tools is compared and the accuracy of the detected text has been improved using post-processing.

The work done in [15] [17] [18] suggests using machine learning models for text classification. However, [16] [19] recommend using deep neural networks for text classification which is an improvement over the traditional machine learning models. Deep neural networks require large training data; therefore, it is advisable to use traditional machine learning models in case training data is less in size. Additionally, [13] suggests using web-based text similarity to retrieve similar texts which is an improvement over classical methods like correlation and TF-IDF proposed in [11] [12] [14]. This thesis initially compares various machine learning models for classifying quoted images, and then proposes a novel web-based text similarity approach to retrieve URLs of extracted text from quoted images and classify the images accordingly.

3.2 Research Objectives

Objectives refer to the tasks or activities used to achieve the goal. This system aims at developing an application which can classify quoted images into containing *Verified* or *Misquoted*. The main objectives of this system are discussed below.

- To acquire quoted image from various sources having different formats followed by pre-processing the images to convert the it into grayscale and remove noise.
- To identify the textual content from the images using the OCR tool and perform post-processing technique to improve the accuracy of the retrieved text.
- To perform a web-based text similarity to obtain the URLs of similar text obtained from the images and verify the extracted domain names against the authentic quotation sites database.
- To identify author's name using Named Entity Recognition and Chunking.
- To classify the quoted images based on the proposed approach.
- To develop a web-based quote verification system to classify quote as *Verified* or *Misquote*.

CHAPTER 4
WORKING OF THE PROPOSED SYSTEM

4.1 Methodology

This section discusses the methodology and technology used to identify text in natural scene images. This method involves the use of OCR tools for text detection and post-processing for the improvement of accuracy. The proposed approach uses scripts written in *Python 3* which runs on *Anaconda software (Jupyter Notebook 4.4.0).*

The block diagram of the proposed system has been presented in Figure 4.1. The various stages involved in this thesis have been explained in detail below. The main aim of the proposed system is to identify text from images followed by post-processing of the detected text. Later, the retrieved text is verified and ultimately the images are classified into their respective class, *i.e., Verified* or *Misquoted.*

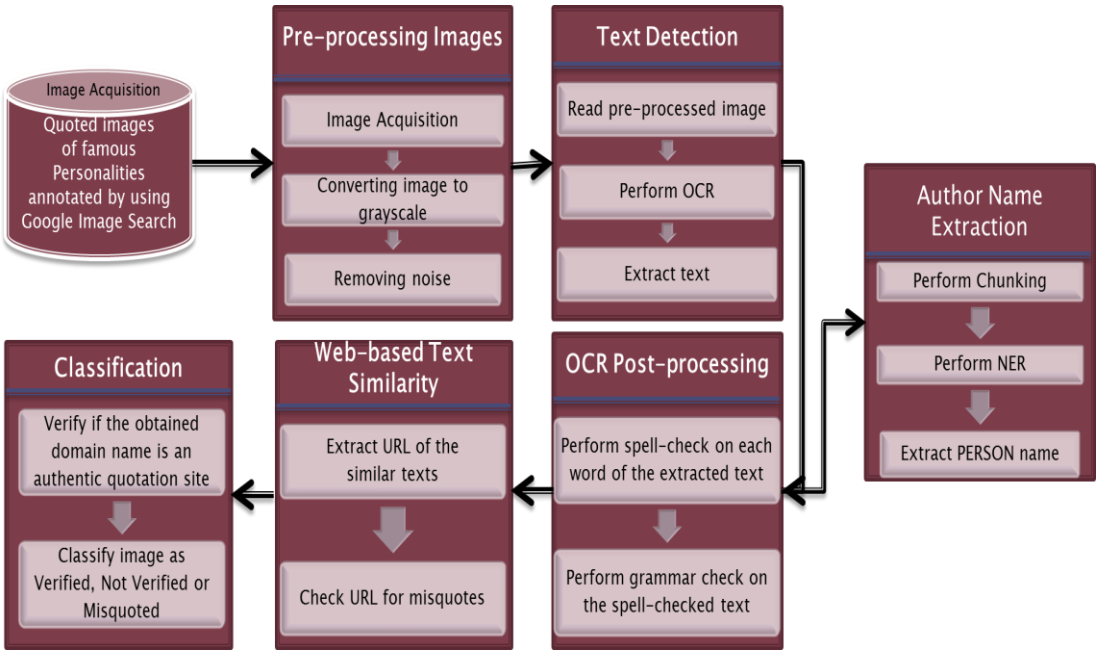


Figure 4.1: Block Diagram of the Proposed Approach [29]

The major stages of the proposed approach have been explained in detail below.

i. ***Image Acquisition:*** The image acquisition phase focus on the methods of obtaining the image from various sources. The *cv2* and *PIL* libraries have been used for reading the images for further processing.

ii. ***Pre-processing image*:** Pre-processing phase helps to enhance the quality of the image for accurate text detection. This includes converting the image into grayscale and removing noise like variation in brightness and color information from images. Non-local Means Denoising algorithm [25] has been used for noise removal from an image and the result obtained can be seen in Figure 4.2 (c). The flow of the preprocessing stage can be seen in Figure 4.2.

(a) (b) (c)

Figure 4.2: Pre-processing stages: (a) Original Image, (b) Converted to grayscale and (c) After Denoising [29]

iii. ***Text Detection*:** Text detection in this experiment has been performed using various OCR tools (like Tesseract-OCR, Google Cloud Vision and AWS rekognition) to extract the text from pre-processed images. Figure 4.3 illustrates the block diagram of the proposed system for OCR tool comparison. For this purpose 3 datasets have been used, namely: IIIT 5K-word [20], KAIST scene text [21] and Kaggle's Handwritten word [22] dataset. All the three datasets combined to form a dataset of around 10k images.

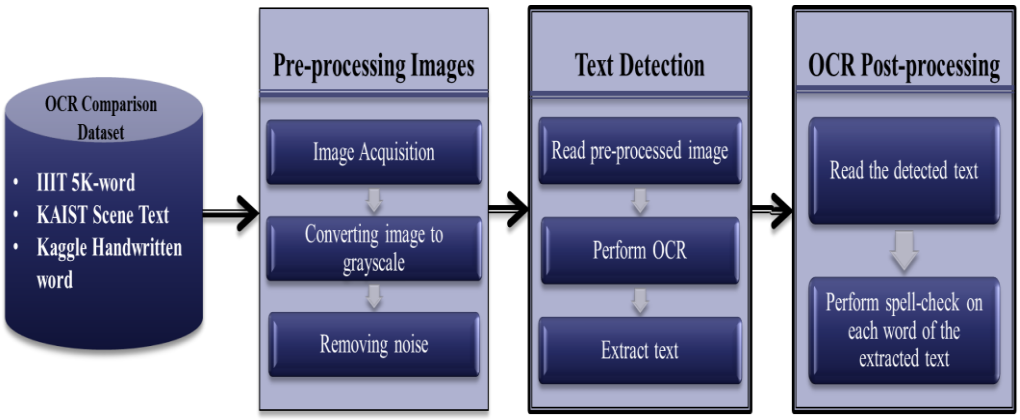


Figure 4.3: Block diagram of OCR tool comparison

The working of each tool has been explained below.

a. *Tesseract-OCR:* Tesseract-OCR is a freely available tool for text detection and retrieval from images. The *pytesseract* library has been used to detect and retrieve the text from the images.

b. *Google Cloud Vision:* The Google Cloud Vision requires Google Cloud Access and creation of an API key for the Cloud Vision project. Moreover, there is a requirement to set environment variables and GOOGLE_APPLICATION_CREDENTIALS which is set as the API key generated. Finally, to use this tool, google-cloud and google-cloud-vision libraries have been used. Then the detection and extraction of the text from images have been performed.

c. *AWS rekognition:* It is a type of OCR tool which requires AWS access. The library boto3 has been used for performing text retrieval form the images. It also requires setting the client as rekognition and the region as us-west-2.

iv. OCR post-processing: After performing OCR, post-processing is done on the retrieved text using automatic spell checking tool to obtain the final result in a machine-readable format. The libraries like autocorrect and *language_check* are available in Python, which helps in correcting the non-word and real-word errors. The *language_check* works on complete sentences and removes spelling and grammatical errors from the retrieved text. On the other hand, autocorrect works on single words and automatically corrects the spelling of the incorrect words. Table 4.1 shows the working of the *autocorrect* and *language_check* tools.

Table 4.1. Working of *autocorrect* and *language_check* tools

Input	**Output**	
	autocorrect tools	***language_check tools***
I an a girl	I an a girl	I am a girl
twu	two	twu
how is you	how is you	how are you
huw	how	huw

It can be summarized that *language_check* performs better in correcting grammatical errors, however *autocorrect* performs better in correcting for individual words which have been misspelled. As the dataset used for OCR tool comparison analysis mainly contains individual words, therefore *autocorrect* has been used. On the other hand, for quoted image verification and classification

initially *autocorrect* is used followed by *language_check* to post-process the retrieved text.

The steps mentioned above helps in text extraction from the images. Figure 4.4 shows the results obtained after each stage of text detection from image using post-processing.

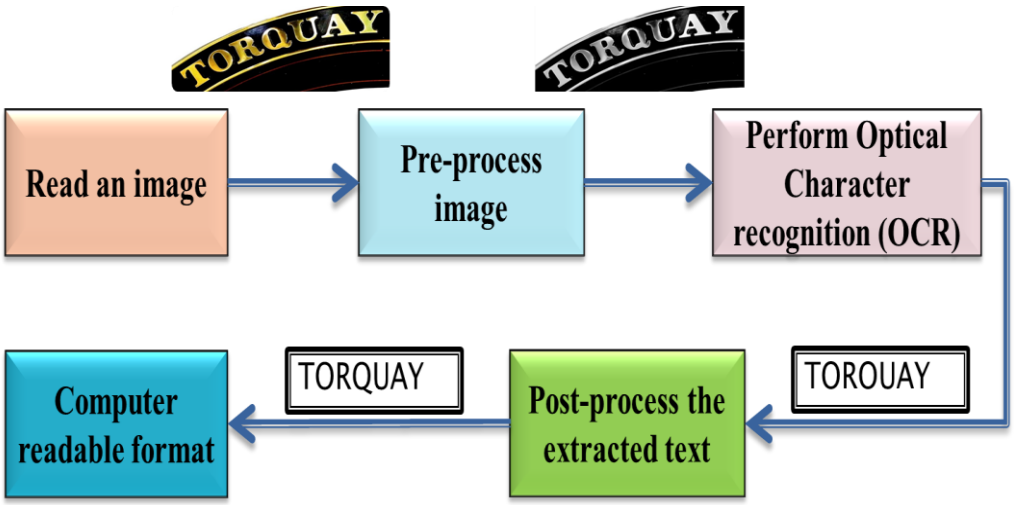
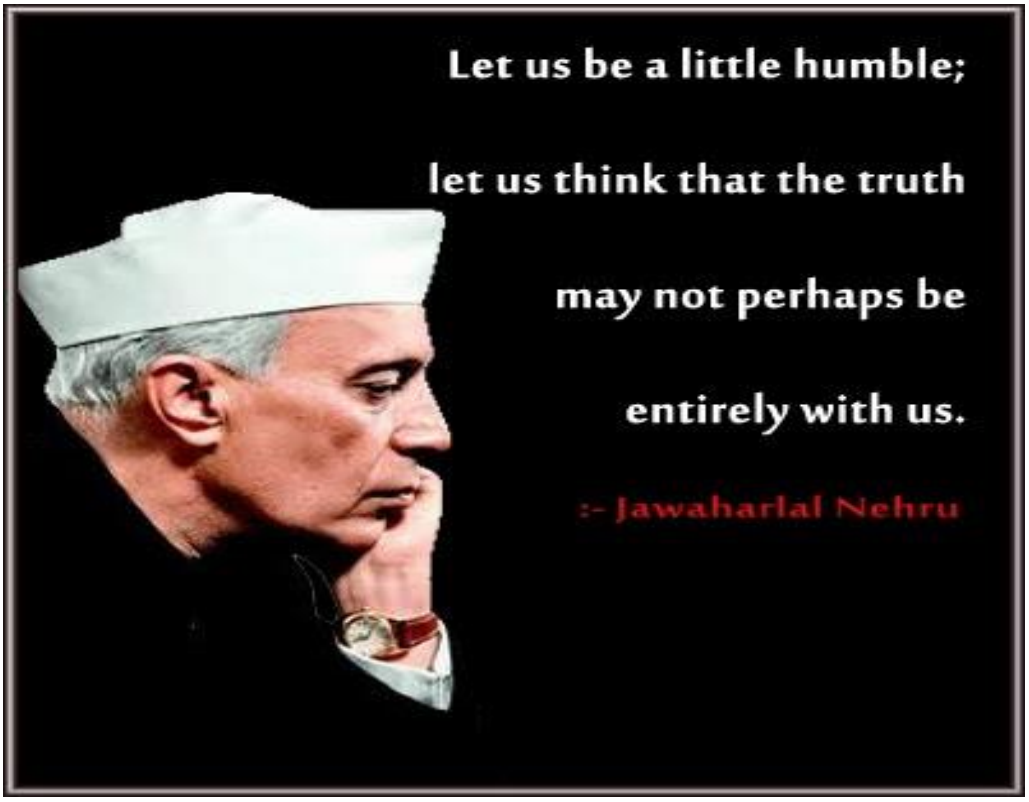
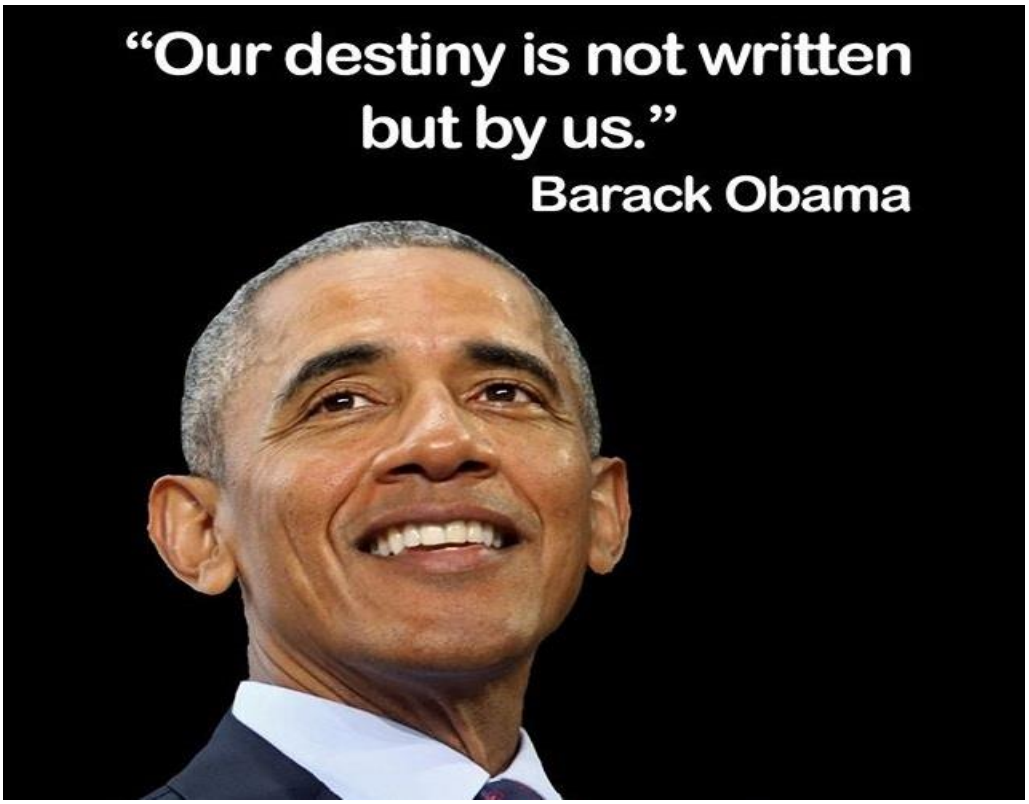


Figure 4.4: Flow diagram of the OCR post-processing [29]

v. ***Author Name Extraction:*** Named Entity Recognition (NER) and Chunking has been performed to identify the author's name from the quoted text. NER is the method to tag the tokens or words which are proper nouns, like name of a person, organization, city, country, *etc.* On the other hand, Chunking is the process of dividing a text into various phrases like Verb (VP), Noun (NP) and Preposition (PP). After implementing NER and Chunking on the post-processed text, the Noun Phrase is identified. Later, the tag PERSON is recognized from the Noun Phrase. This identified person's name is extracted as the Author's name. For example, for Figure 4.5 (a) author's name extracted is *Jawaharlal Nehru*. Similarly, for Figure 4.5 (b) extracted author's name is *Barack Obama*.



(a) Quote by Jawaharlal Nehru



(b)Quote by Barack Obama

Figure 4.5: Sample images from the dataset [29]

vi. ***Web-based Text Similarity:*** To extract the URLs, a web-based text similarity of the retrieved text has been performed. For this purpose, *Google Custom Search* engine has been used. The retrieved text is provided as search input to Google Custom Search engine and the top 10 results, *i.e.*, URLs are retrieved. Figure 4.6 shows the working of Google Custom Search Engine. In this example, a misquote by Abraham Lincoln has been provided as input of search result. The misquote is "*In the end, it's not the years in your life that count. It's the life in your years".* The extracted URLs can be seen in Figure 4.6. Also, it is visible that 5th URL contains a term *fake*, therefore this image is classified as *Misquoted.*

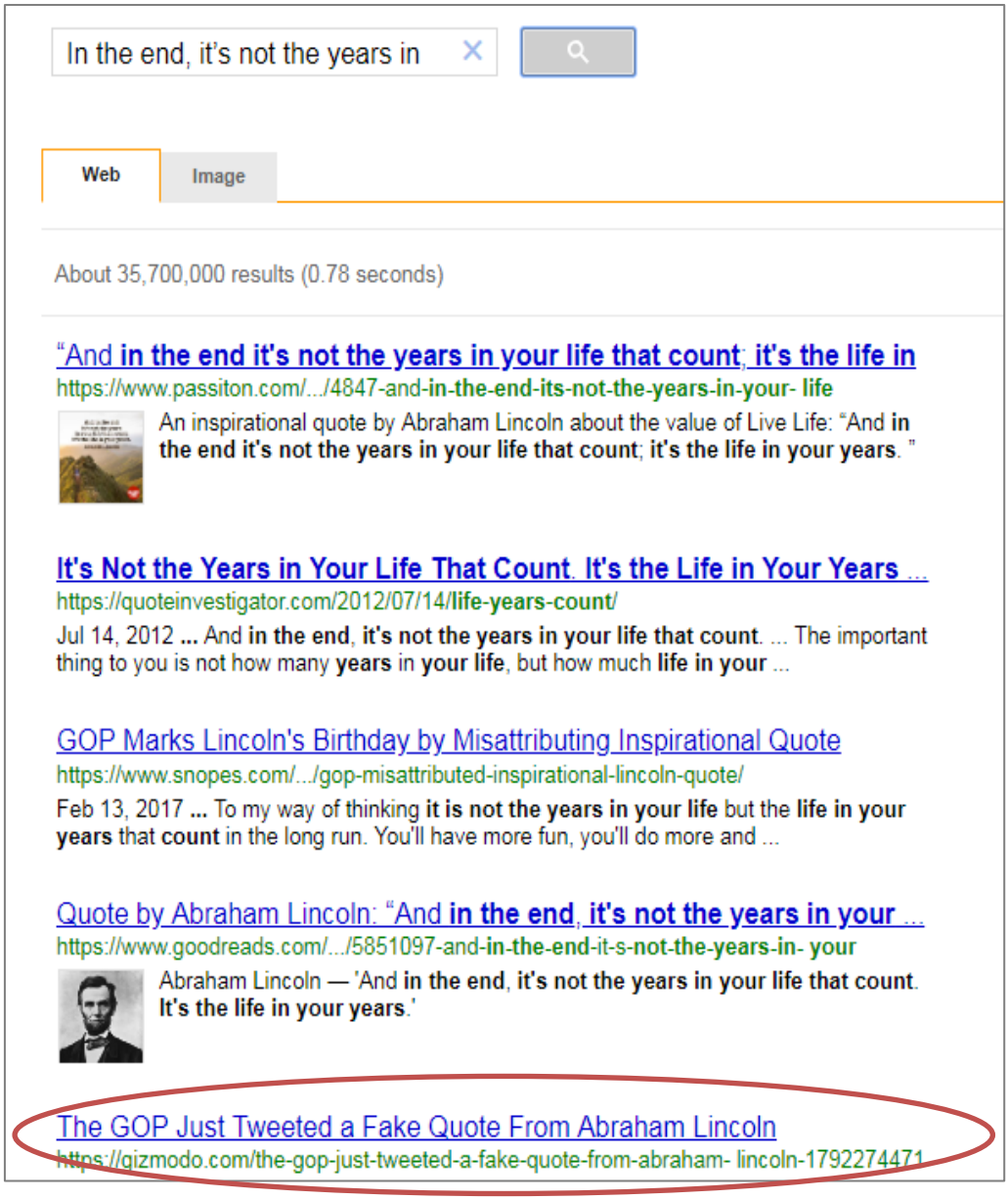


Figure 4.6: A sample search result of Google Custom Search Engine

On extracting top 10 URLs using Google Custom Search Engine, there is a need to check for the presence of fake terms like *fake, misquote, misattributed, etc.* Additionally, domain name is extracted from the URLs.

vii. ***Classification:*** Once if the URLs do not contain information like *fake, misquote, misattributed, etc.,* then there is a need to check the authenticity of the retrieved domain names. For this purpose, a database of authentic site names has been created. The database created contains 100 quotation sites. Table 4.2 shows the top 20 quotation sites from the created database.

Table 4.2. Top 20 quotation sites from the created database

S.No.	Quotation Sites
1	www.brainyquote.com
2	www.quotery.com
3	www.great-quotes.com
4	www.searchquotes.com
5	www.quotesdaddy.com
6	en.wikiquote.org
7	www.worldofquotes.com
8	quotelicious.com.*.com
9	www.quotegarden.com
10	www.goodreads.com
11	www.movemequotes.com
12	www.famousquotesandauthors.com
13	www.quotexo.com
14	whitelight.social
15	medium.com
16	www.telegraph.co.uk
17	www.forbes.com
18	www.amusingquotes.com
19	www.famous-quotes.com
20	www.coolquotes.com

According to above verification results, there is a need to classify the image into *Verified* or *Misquoted*. If the retrieved URLs contains terms like fake, misquote, misattributed, *etc.,* then the image is classified as Misquoted. Also, in case the obtained domain names are not in the list of authentic sites, then also the images are classified as *Misquoted*. Further, in case the no URL is obtained through Google Custom Search result or the obtained domain names do not belong to authentic sites, then the image is classified as missed sample. Ultimately, if the

obtained domain names are from the list of authentic sites, then the image is classified as *Verified*. The classification of images is based on the rules mentioned in Table 4.3 where 0 means *No* and 1 means *Yes*.

Table 4.3. Rules for Classification

Presence of Fake Terms in URL	**Authentic Domain Name**	**Class**
0	0	*No Class*
0	1	*Verified*
1	0	*Misquoted*
1	1	*Misquoted*

In case any sample is classified as *No Class*, then it will be considered as a missed sample. In this thesis, a comparison of the proposed approach with the traditional machine learning models, *i.e.*, Logistic Regression, Naïve Bayes and SVM have also been discussed.

4.2 Dataset Collection

As the proposed approach has two tasks, therefore, different datasets have been used for each task. Section 4.2.1 discusses dataset used for comparing performance of the OCR tools and Section 4.2.2 discusses datasets used for verification and classification of the quoted images.

4.2.1 Dataset Used for OCR tool comparison

The proposed approach uses a combined dataset of IIIT 5K-word dataset [20], KAIST Scene Text Dataset [21] and Handwritten word dataset (Kaggle) [22]. The *IIIT 5K-word dataset* contains Google search images. These collected images contain words appearing in billboards, signboard, house numbers, house name plates and movie posters. The dataset includes 5000 cropped word images from scene texts, word processing documents, spreadsheets, and images produced by digital cameras. A sample of this dataset is shown in Figure 4.7 (a) which is a cropped image of the word *HOLLYWOOD*.

The *KAIST scene text dataset* is a collection of 3000 images clicked in several lighting scenarios like sunny day, night time, under artificial lights, *etc.* The images in this dataset contain few high-resolution digital camera images and few low-resolution

mobile phone camera images which are resized to 640x480. The images are of various billboards written in 2 languages, Korean and English. The images of the English language have been used for this experiment. A sample image of the dataset is Figure 4.7 (b).

The *Handwritten word dataset* is freely available on Kaggle and contains images of six English language words in both upper and lowercase alphabets. Each image is a word written by different people. The dataset comprises 4253 text images for handwriting recognition containing 2513 lowercase words and 1740 uppercase words. Figure 4.7 (c) is a sample showing a handwritten image of the word BOXING.

(a) IIIT 5K-word [20]

(b) KAIST scene text [21]

(c) Kaggle Handwritten word [22]

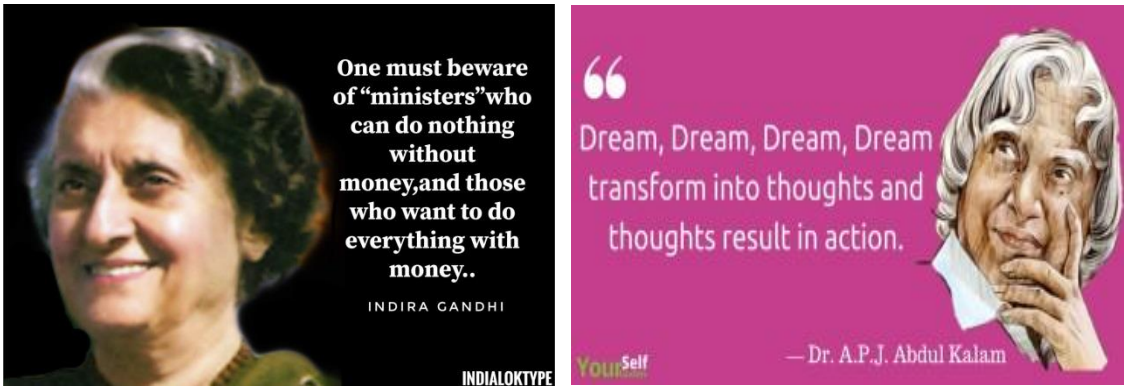
Figure 4.7: Sample images from the datasets (a), (b) and (c)

As discussed in this section, the dataset used in this approach consists of around 10k data of images. The images have many variations in terms of textual content, quality and format of images, the mode used for capturing the images, *etc.* On each image,

OCR is performed for text detection followed by post-processing to enhance the efficiency of the recognized text.

4.2.2 Dataset Used for Classification

The dataset used in this experiment is an annotated dataset which is a combination of quoted images of famous personalities. Figure 4.8 shows some sample images from the dataset used for the purpose of image classification. The mentioned dataset is used as the testing data to evaluate the performance of the classification models. It is an annotated dataset of 3237 labelled images containing famous quotations collected through google images. The dataset consists of 2735 images containing verified quotes and 502 images containing misquotes**.**



(a) Quote by Indira Gandhi (b) Quote by Dr. A. P. J. Abdul Kalam

Figure 4.8: Sample images from the testing dataset [29]

The traditional classification approaches such as Logistic Regression, Naïve Bayes and SVM needs training data to create a machine learning model. For this purpose, a training dataset has been created which consists of 7225 quotes having two fields, *i.e.*, "*Text"* which contains the quote and *"Class"* which is the label assigned to the quote. *Class 1* means *Verified* quote and *Class 0* means *Misquote*. There are total 7001 verified quotes and 224 misquotes available in this dataset. Figure 4.9 shows sample of the training dataset.

	Text	Class
2	640k ought to be enough for anyone.	0
3	A 2001 survey of business owners with MBAs conducted by the Rochester Institute	1
4	A bank is a place where they lend you an umbrella in fair weather and ask for it ba	1
5	A banker is a fellow who lends you his umbrella when the sun is shining but wants	0
6	A battle lost or won is easily described, understood, and appreciated, but the mor	1
7	A beautiful thing never gives so much pain as does failing to hear and see it.	1
8	A Bill of Rights is what the people are entitled to against every government, and v	1
9	A birthday is just another day where you go to work and people give you love. Age	1
10	A budget tells us what we can't afford, but it doesn't keep us from buying it.	1

Figure 4.9: Sample of the training dataset [29]

The Table 4.4 summarizes the datasets used for the proposed approach. It describes the dataset name, number of items present in the dataset and the task it has been used for in the thesis

Table 4.4. Datasets used for the proposed approach

Name of the Dataset	Number of Items	Task
IIIT 5K-Word	5000	Comparison of OCR tools
KAIST scene text (English words)	762	Comparison of OCR tools
Kaggle Handwritten word	4253	Comparison of OCR tools
Annotated dataset of quoted images(testing dataset)	3237	Proposed system as the testing dataset
Quotations (training dataset)	7225	Training machine learning models

.

In this chapter, the methodology used for the proposed system has been explained in details. The datasets used for implementing the proposed approach have also been discussed in this chapter. The implementation and design specification of the proposed system have been documented in Chapter 5.

CHAPTER 5
IMPLEMENTATION AND DESIGN SPECIFICATION

This chapter highlights the design and implementation of the *Quote Examiner* system. It is a web-based system which browses quoted images and extracts its information followed by classification of the image into categories *Verified* or *Misquoted.*

5.1 Work Breakdown Structure

Figure 5.1 depicts the work breakdown structure of the project. This project comprises mainly of two components, *i.e.*, Front End and Back End. The flow of the project task has been diagrammatically represented in Figure 5.1.

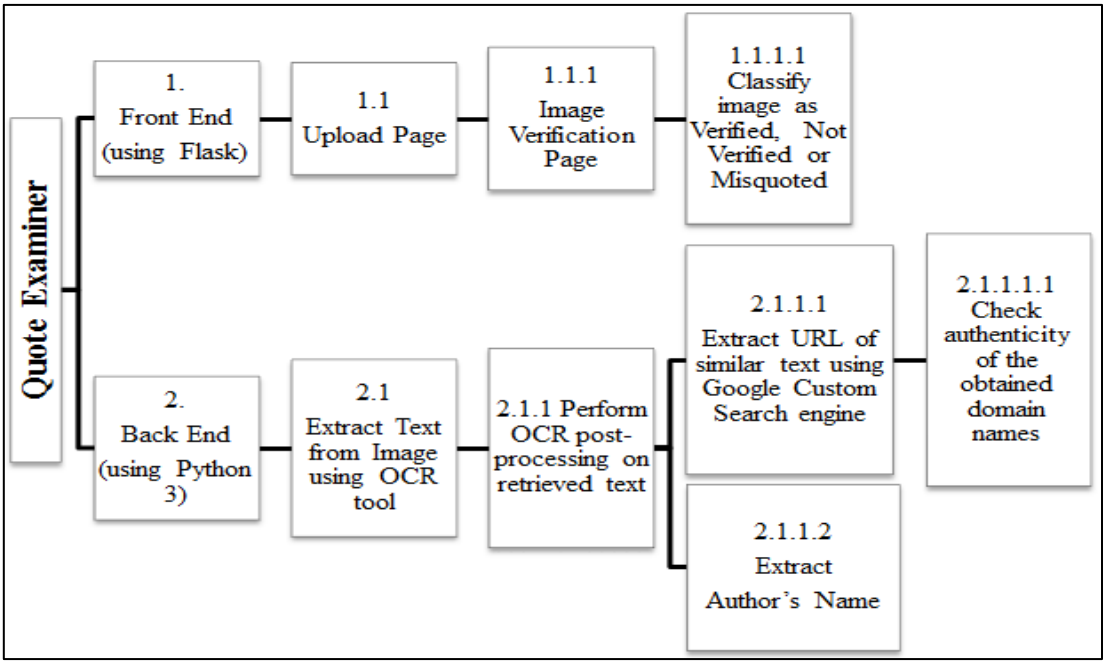


Figure 5.1: Work Breakdown Structure of the proposed system [29]

The work breakdown structure of the Quote Examiner website is given as below.

1. Front End (using Flask)

    1.1 Upload Page

        1.1.1 Image Verification Page

            1.1.1.1 Classify image as *Verified or Misquoted*

2. Back End (using *Python 3*)

2.1 Extract Text from Image using the OCR tool

2.1.1 Perform OCR post-processing of retrieved text

2.1.1.1 Extract URL of similar text using Google Custom Search Engine

2.1.1.1.1 Check the authenticity of the obtained domain names

2.1.1.2 Extract Author's Name

5.2 System Components

There are various system components at work in the proposed system, each playing a specific role in implementation. Together, they allow the system or application to function efficiently and accurately. The two main components of this system are back end and front end which are discussed below.

- **Back End/Development**

The server-side is often referred to as the back end of the system. Usually web developers work on the back end. This basically attributes to the fact that how the site works, changes and gets updated.

- ***Language Used:***

  - *Python 3* - A general-purpose interactive, object-oriented, interpreted, and high-level programming language.

- ***Platform:***

  - *Anaconda Enterprise* - An enterprise-ready, scalable and secure data science platform that helps in empowering teams to govern the assets of data science, collaborate and deploy projects in data science.
  - *Jupyter Notebook (4.4.0)* - Anaconda provides an open-source web application known as the Jupyter notebook that allows its users to share and create documents that consist of live code, visualizations, equations, and narrative text. This tool can be utilized for data cleaning and transformation, simulation, data visualization, statistical modeling, machine learning, *etc.*
  - *MS Excel* - Microsoft Excel is a spreadsheet created by Microsoft for Windows, Android, MacOS, and iOS. It includes calculations, pivot tables, graphing tools,

and a macro programming language known as Visual Basic for Applications. The dataset used for this experiment has been stored in MS Excel sheets.

- ***Libraries:*** The following Python libraries have been used for various purposes in the proposed system.

  - *PIL and cv2*– Used for reading images and processing them.
  - *pytesseract* – Tesseract-OCR tool which is required for extracting text from image.
  - *google-cloud-vision* – The library used for performing text detection from images using Google Cloud Vision.
  - *boto3* – The Python package used for identifying text from images using AWS rekognition.
  - *nltk* – Required for performing NLP tasks like tokenization, POS tagging, NER, chunking, *etc.*
  - *language-check and autocorrect* – Libraries used for correcting spelling errors and grammatical errors in any text.
  - *customsearch api* – The Pyhton API for performing Google search using a Custom Search Engine.

- **Front End/User-Interface**

The interface has been designed using Python 3 on Anaconda platform and run on localhost. The following components have been used separately for designing the interface.

- ***Language*:**

  - *Hypertext Markup Language* (*HTML)* – It is the standard markup language for creating various web applications and web pages.
  - *JQuery* - It is a JavaScript library which has been designed to simplify HTML DOM tree traversal and manipulation, including animation, event handling, and Ajax.

- ***Interface:*** *Flask* is a micro web framework which has been written in Python as it does not need any particular tools or libraries. For using Flask in this system, the library *flask* has been used.

5.3 Flowchart of the proposed system

Figure 5.2 represents the workflow of this project. It explains all the steps involved in the proposed approach. The flowchart involves reading the input image, recognizing text from image using Google Cloud Vision followed by post-processing, *i.e.*, removal of linguistic errors. Next step is to identify author's name and verification of the quoted images.

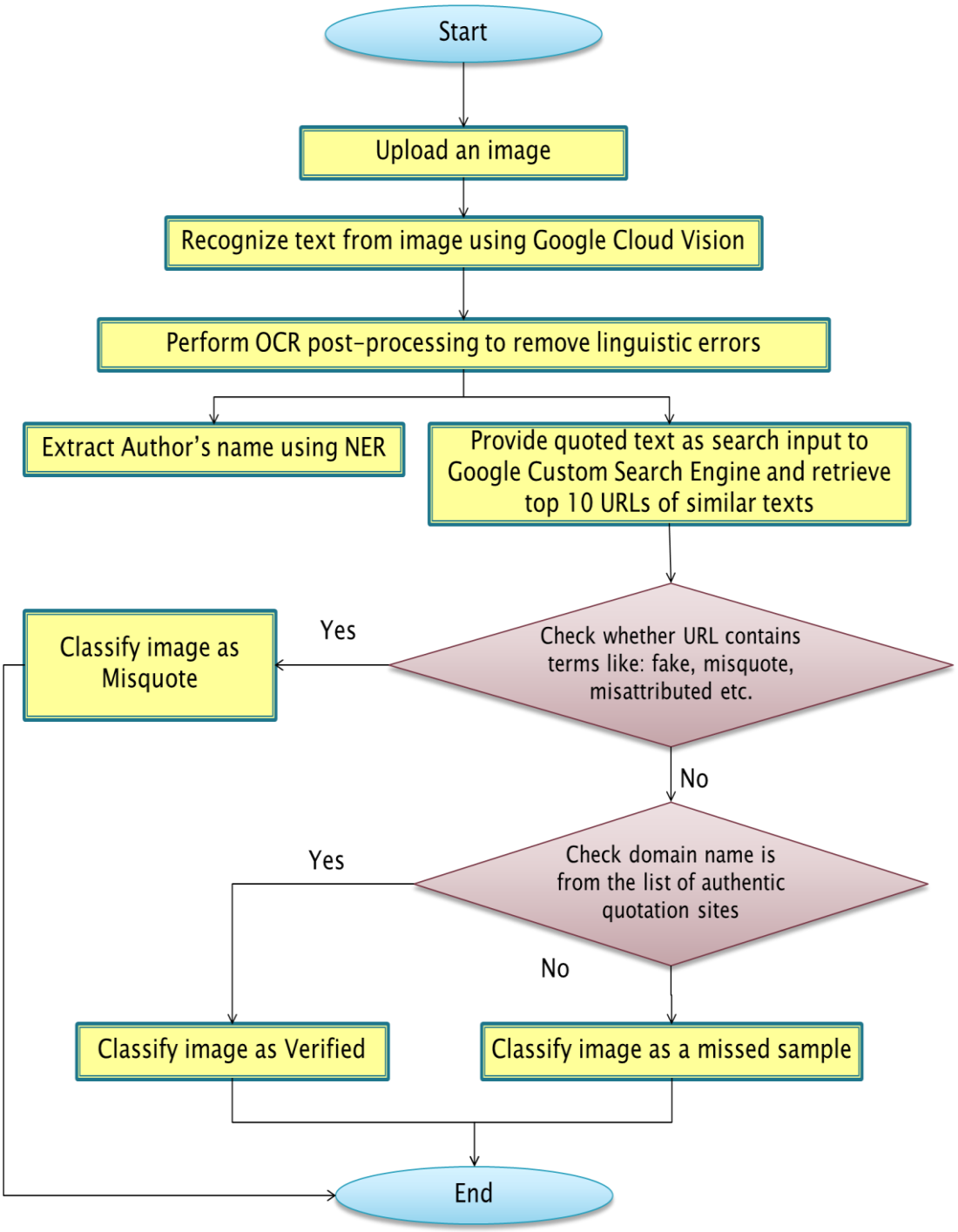


Figure 5.2: Flowchart of the proposed system [29]

5.4 User Interface Diagrams

User Interface diagrams are custom diagrams used to view a system's user interface. Figure 5.3 represents the User Interface diagram for the Image Verification system.

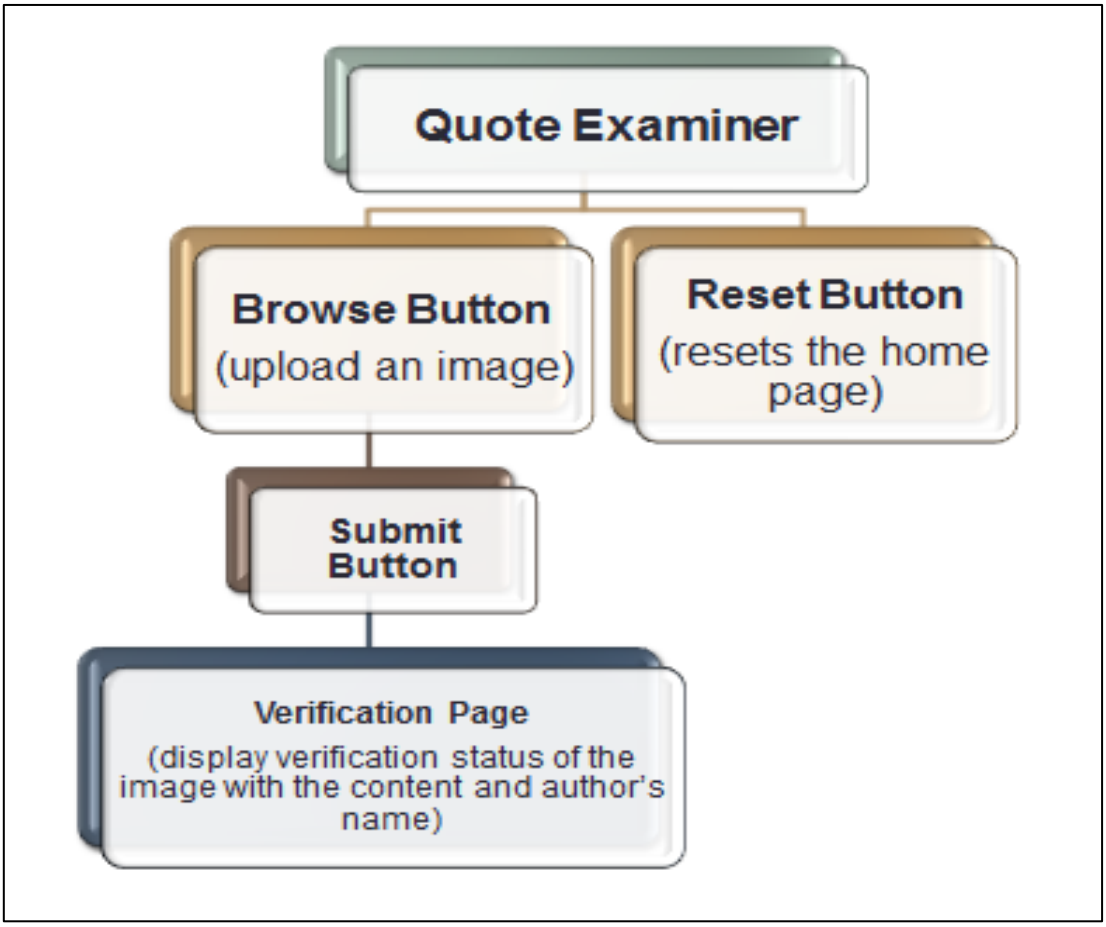


Figure 5.3: User Interface diagram of Image Verification system [29]

The flow of the application or User interface is explained below.

- The user opens the Quote Examiner Home page.
- Browse the quoted image to be verified and performs either of the two actions, reset the page using Reset button or submit the image using the Submit button.
- On submitting the User is able to verify the status of the image, whether it contains *Verified*, *Misquoted* or *Not a Verified* quote.

5.5 Snapshots of the Working Model

The web-based interface has been designed using *Python 3* on *Anaconda* platform. *Flask* library has been used to design the interface for this system. The system reads an input image, retrieves the quote and author name and classifies the image as

containing *Verified quote* or *Misquote*. The Figure 5.4 to 5.8 shows the working of this system. Below are the steps highlighting the implementation of the proposed system.

- *Step 1: HOME/UPLOAD PAGE*

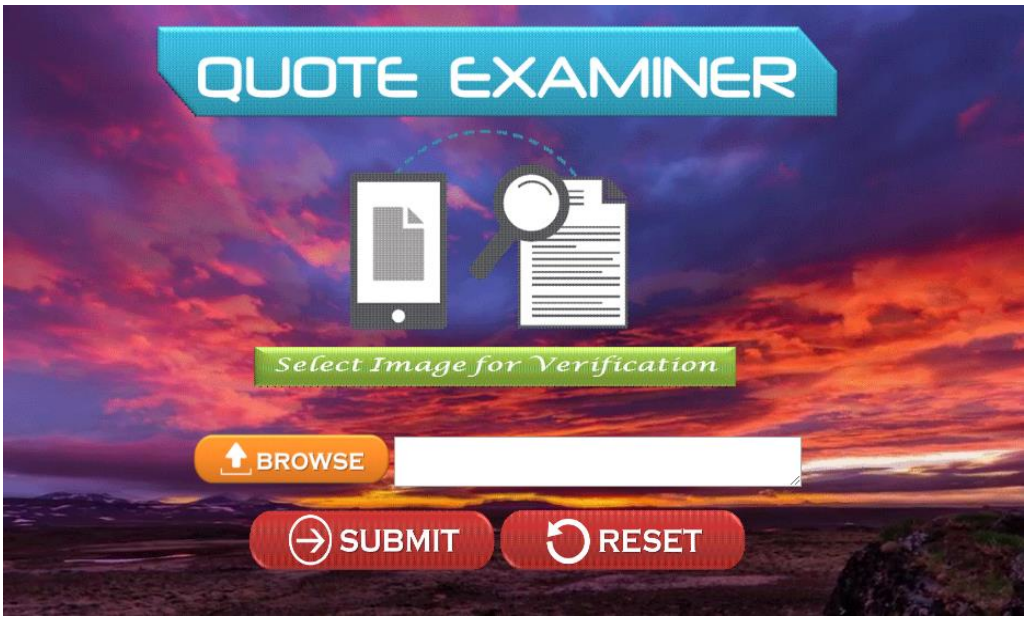


Figure 5.4: Screenshot of HOME PAGE [29]

- *Step 2: Using BROWSE BUTTON*

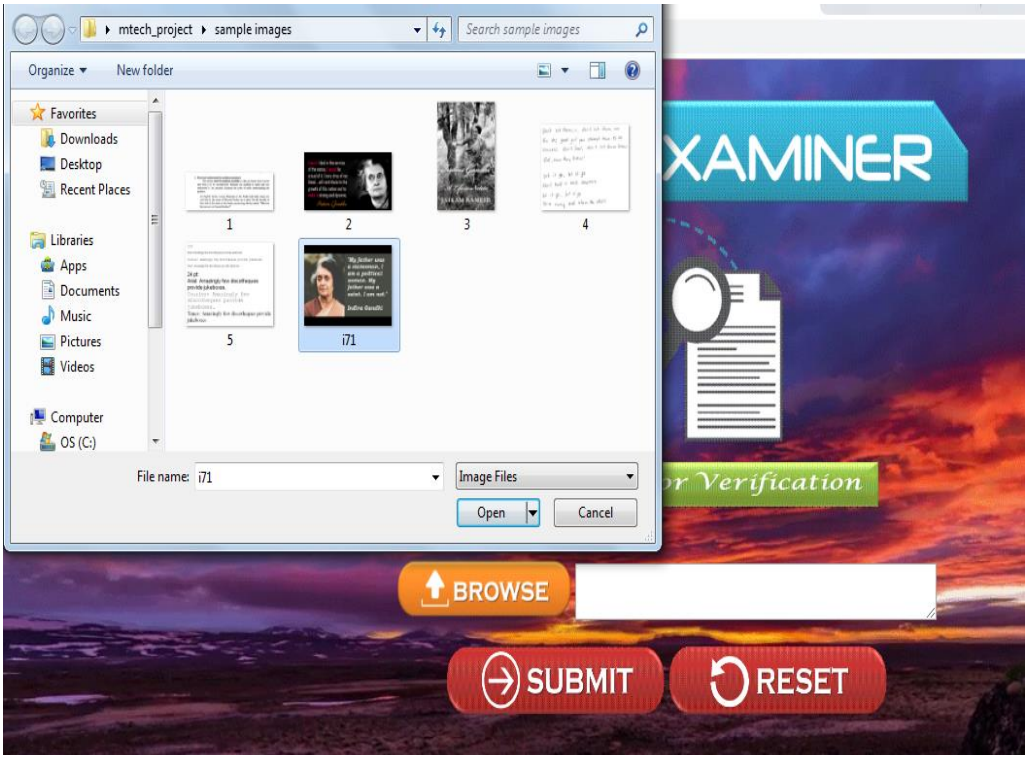


Figure 5.5: Screenshot of browsing an image [29]

- *Step 3: Using RESET BUTTON*

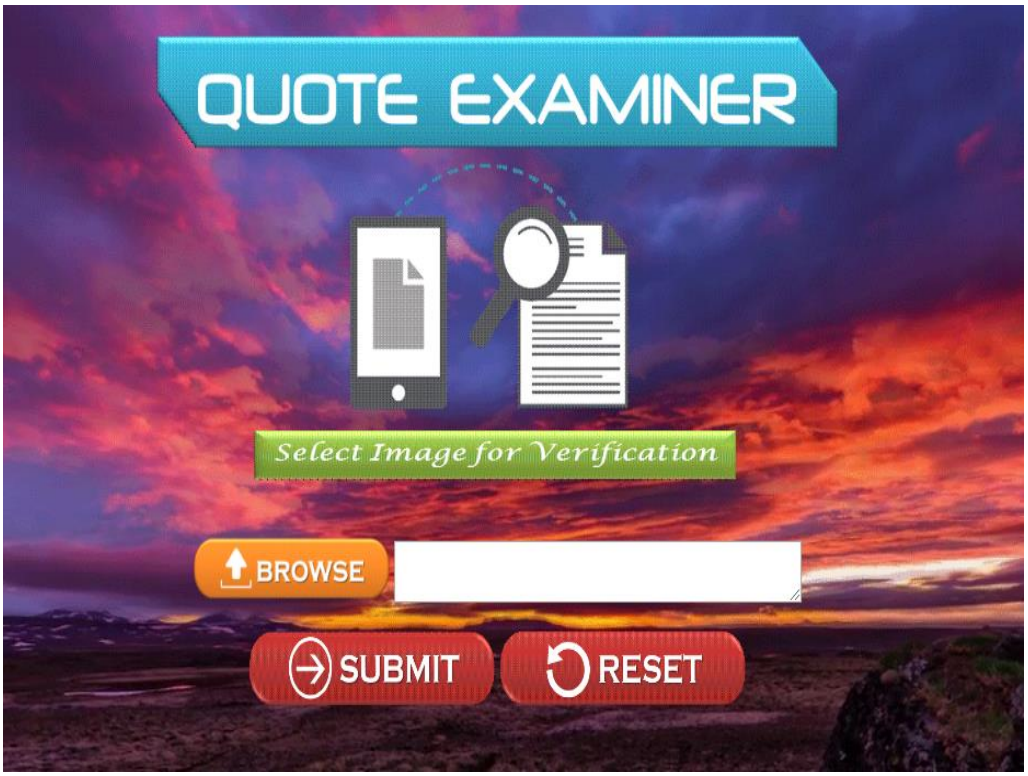


Figure 5.6: Screenshot after pressing the RESET button [29]

- *Step 4: Using SUBMIT BUTTON*

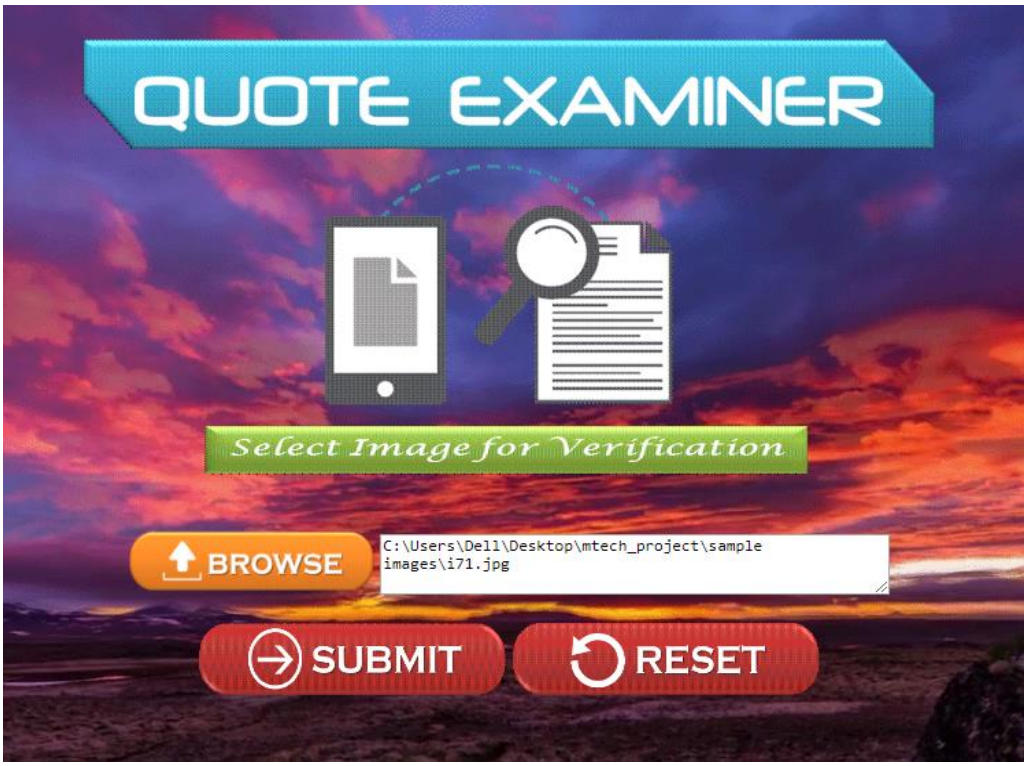
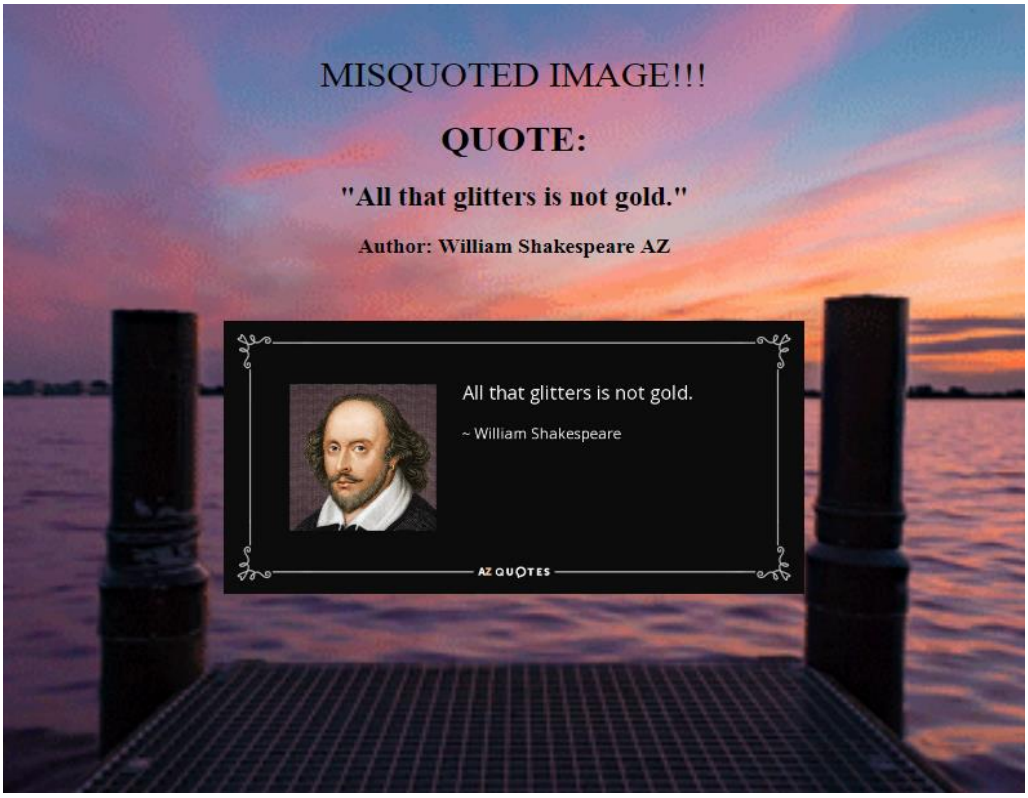


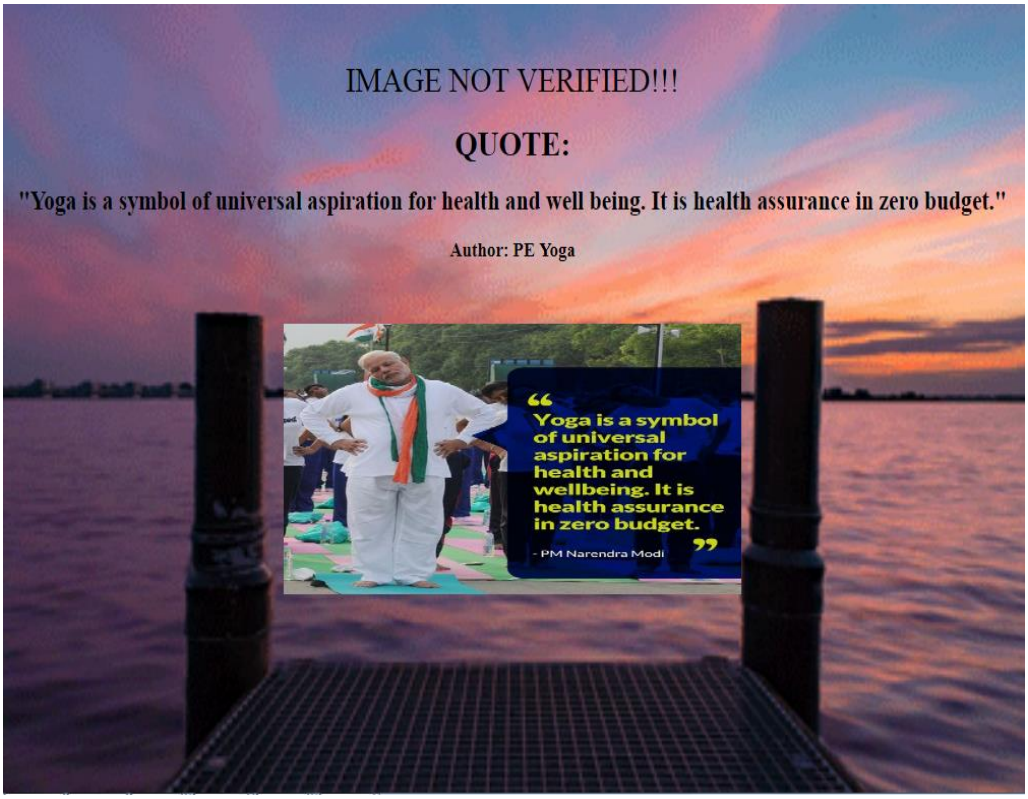
Figure 5.7: Screenshot after image browsing (path of the image appears) [29]

- *Step 5: IMAGE VERIFICATION PAGE*

(a) In case image contains Verified Quote



(b)In case image contains Misquotes



(c) In case image does not contain Verified Quote or a Misquote

Figure 5.8: Result obtained through the proposed system [29]

This chapter discusses various techniques implemented in the proposed approach. The proposed approach has been performed in two parts, first is the comparison of OCR tools for text detection from image and second is verification and classification of the quoted images. The obtained results have been discussed in chapter 6.

CHAPTER 6
RESULTS AND DISCUSSION

This chapter focuses on the results obtained by implementing the proposed system. Section 6.1 highlights the performance analysis of the OCR tools, *i.e.,* Tesseract-OCR, Google Cloud Vision and AWS rekognition in detecting text from images have been highlighted. Further, Section 6.2 discusses the results obtained after implementing the proposed approach of classifying the quoted images into the categories of *Verified* and *Misquoted* using Google Cloud Vision and web-based text similarity. Moreover, the performance of traditional machine learning models for classifying quoted images has also been presented in Section 6.2. Additionally, a detailed comparision analysis of the traditional machine learning models and the proposed approach has been highlighted in Section 6.3.

6.1 Comparison of the OCR tools

There are various OCR tools available which help in converting visual data into editable textual documents. None of the available OCR tools are perfect in extracting information from the images accurately, therefore a method of post-processing on the retrieved text to improve the accuracy of the detected text from images has been proposed. Hence, a performance analysis between various OCR tools like Tesseract-OCR, Google Cloud Vision and AWS rekognition is performed on natural scene images.

6.1.1 Performance Analysis of OCR Tools

The experimental results have been presented in this section. For any given image, each tool returned the text detected from it. Further, the detected text has been processed for removing spelling errors to increase the accuracy and overall performance of the tool. In case the tool is unable to detect any text from the image, then an empty response is returned. Table 6.1 shows few sample cases where the tools were either able to recognize the entire text of the image correctly or not at all. To analyze the performance of each tool, performance metrics have been calculated which are mentioned in Section 6.1.2 and the observed results have been explained in Section 6.1.3.

Table 6.1. The ability of each tool to recognize text from the given sample images

Sample Images								
Tesseract-OCR	×	✓	×	✓	✓	×	✓	×
Google Cloud Vision	×	×	✓	×	✓	✓	✓	×
AWS rekognition	✓	×	×	✓	×	✓	✓	×

6.1.2 Performance Metrics of OCR Tools Comparison

For analyzing the performance of the OCR tools, there is a need to evaluate the accuracy of each tool in recognizing the text from an image. Besides accuracy, precision and recall are highly helpful in analyzing the performance of each tool. Before analyzing the results, there is a need to understand the meaning and use of these metrics. These metrics have been discussed below, where *T represents total images in the dataset, C represents correctly identified images,* and *I represent incorrectly identified image.*

a. *Accuracy: It is the percent of correct predictions,* in this experiment; it is the percent of data extracting correct text from the entire dataset.

$$Accuracy = C / T \quad (6.1)$$

b. *Recall: It indicates the percent of positive cases caught, i.e.,* percent of data where text was retrieved from images, *i.e.,* all the cases where text was recognized correctly along with incorrectly recognized cases.

$$Recall = (C + I) / T \quad (6.2)$$

c. *Precision: It is the percent of correct predictions (correctly identified text of the images) from the positive cases (images from which some text has been extracted), i.e.,* percent of data from the retrieved cases where the retrieved text was recognized correctly.

$$Precision = C / (C+I) \quad (6.3)$$

d. *F1 measure:* To achieve an optimal combination of precision and recall, a metric called *F1 measure is used which is a harmonic mean of the two metrics.*

$$F1\ measure = (2 * (4) * (5)) / ((4) + (5)) \quad (6.4)$$

6.1.3 Results and Observation of OCR Tools Comparison

The experimental results obtained by performing the proposed approach have been documented in Table 6.2, 6.3 and 6.4. In this experiment, firstly, the results have been calculated for each tool without using post-processing and also the results have been obtained after post-processing step, *i.e.,* correcting the misspelled words. In the Table 6.2, 6.3 and 6.4, metrics are abbreviated as below.

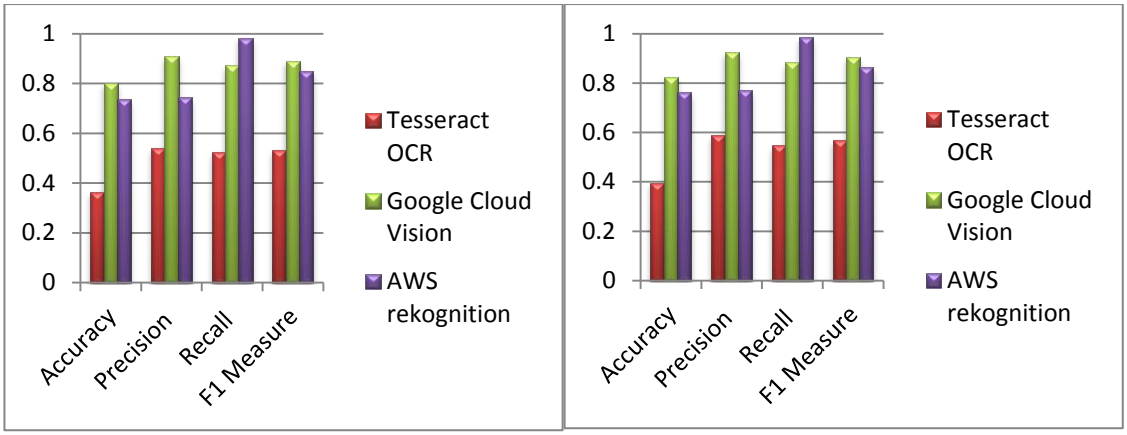
- Correctly identified text as C, Incorrectly identified text as I, and No text identified in the images as N
- Accuracy as A, Precision as P, Recall as R, and F1-measure as F1.

It has been observed from the results obtained in Table 6.2 that Google Cloud Vision [23] has the highest accuracy in recognizing text from the IIIT 5K-Word dataset [20] and also a high value of precision and recall which indicates that this tool performs well on the images of this dataset. The Tesseract-OCR [2] shows the least accuracy, precision and recall indicating a low performance in detecting text from the images.

Table 6.2. The results obtained after performing OCR and post-processing of IIIT 5K-Word Dataset containing a total of 5000 images

Tools	C	I	N	A	P	R	F1
Tesseract OCR	1819	1540	1641	36.38%	54.15%	52.57%	53.35%
After Post-Processing	1978	1381	1641	39.56%	58.89%	54.65%	56.69%
Google Cloud Vision	4013	409	578	80.26%	90.75%	87.41%	89.05%
After Post-Processing	4121	340	539	82.42%	92.37%	88.43%	90.36%
AWS rekognition	3683	1257	60	73.66%	74.55%	98.39%	84.83%
After Post-Processing	3811	1129	60	76.22%	77.14%	98.45%	86.50%

It has also been observed from Figure 6.1 that the accuracy in recognizing the text from images has been significantly increased after using post-processing on the output retrieved from the tools.



(a) Without OCR post-processing (b) After using OCR post-processing

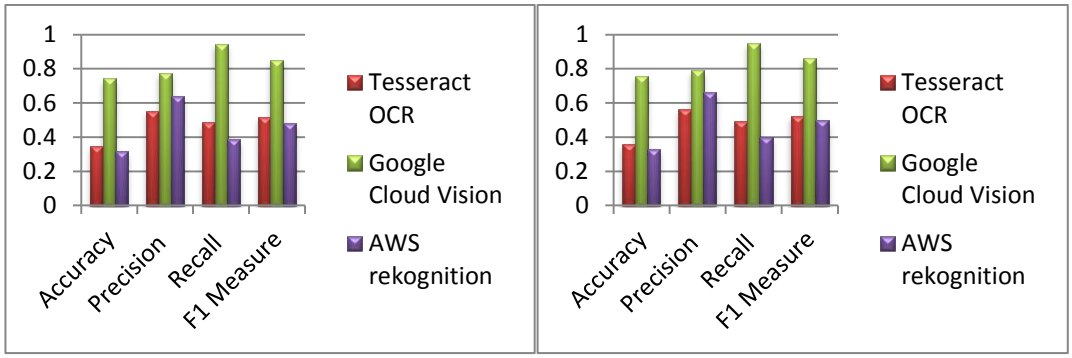
Figure 6.1: Performance analysis of OCR tools using IIIT 5K-Word dataset [29]

The results obtained in Table 6.3 indicate that Google Cloud Vision [23] outperforms the other two tools in recognizing text from the KAIST scene text dataset [21] whereas, AWS rekognition [24] achieves the least accuracy and a very low recall which indicates that this tool was not able to recognize text from a large portion of the dataset. This dataset contained *.bmp* images which were not identified by AWS rekognition [24] making its performance in recognizing text significantly poor. Again, Tesseract-OCR [2] could not perform well in this case even.

Table 6.3.The results obtained after performing OCR and post-processing of KAIST Scene Text Dataset containing a total of 762 images

Tools	C	I	N	A	P	R	F1
Tesseract OCR	266	215	281	34.91%	55.30%	48.63%	51.75%
After Post-Processing	271	210	281	35.56%	56.34%	49.09%	52.47%
Google Cloud Vision	566	164	32	74.28%	77.53%	94.65%	85.24%
After Post-Processing	578	152	32	75.85%	79.18%	94.75%	86.27%
AWS rekognition	242	137	383	31.76%	63.85%	38.72%	48.20%
After Post-Processing	252	127	383	33.07%	66.49%	39.68%	49.70%

Figure 6.2 highlights that using post-processing on the output retrieved from the tools increased the calculated accuracy, but not to a great extent as the images in consideration contained few Korean names which could not be automatically corrected using the proposed post-processing method as it deals with only English words.



(a) Without OCR post-processing (b) After using OCR post-processing

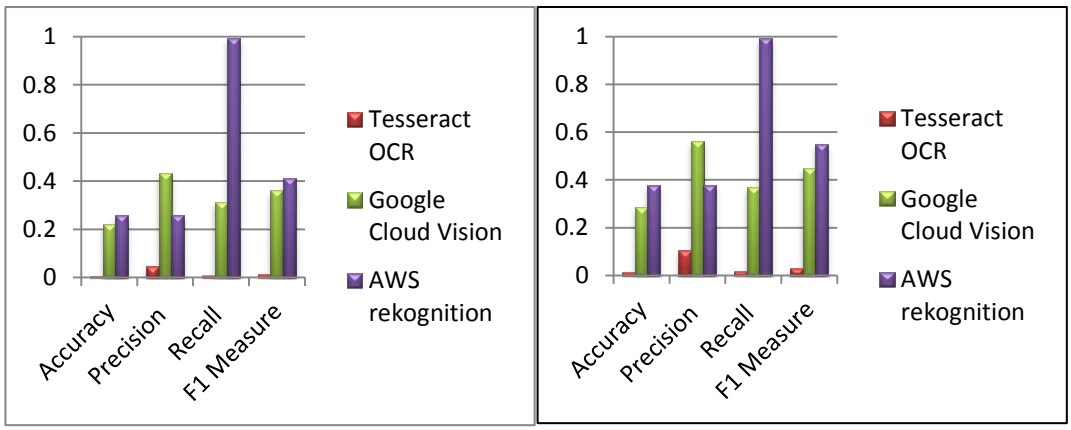
Figure 6.2: Performance analysis of OCR tools using KAIST Scene text dataset [29]

The results obtained in Table 6.4 shows that AWS rekognition tool [24] performs best in recognizing text from the Handwritten Word dataset [22]. The Google Cloud Vision [23] and Tesseract-OCR [2] tools were not able to recognize any text from a large part of the dataset. AWS rekognition [24] performed better than the other tools in identifying text from handwritten images; however, the results obtained were not appreciable.

Table 6.4. The results obtained after performing OCR and post-processing of Handwritten Word Dataset containing a total of 4253 images

Tools	C	I	N	A	P	R	F1
Tesseract OCR	266	215	281	34.91%	55.30%	48.63%	51.75%
After Post-Processing	271	210	281	35.56%	56.34%	49.09%	52.47%
Google Cloud Vision	566	164	32	74.28%	77.53%	94.65%	85.24%
After Post-Processing	578	152	32	75.85%	79.18%	94.75%	86.27%
AWS rekognition	242	137	383	31.76%	63.85%	38.72%	48.20%
After Post-Processing	252	127	383	33.07%	66.49%	39.68%	49.70%

It is visible from Figure 6.3 that AWS rekognition [24] has a very low precision value but a very high recall value. This is due to the reason that the tool is able to recognize the relevant text efficiently, but it also misrecognizes a lot of bleed-through text, *i.e.*, irrelevant marks as characters leading to a low precision value. Despite the poor performance in recognizing text from images, the highest improvement in accuracy by using post-processing method has been achieved in the case of the handwritten dataset [22]. The success of the post-processing method is due to the presence of a lot of misspelled words in the retrieved text. All these spelling errors were corrected using the proposed post-processing method.



(a) Without OCR post-processing (b) After using OCR post-processing

Figure 6.3: Performance analysis of OCR tools using Handwritten Word dataset [29]

The derived results show that Google Cloud Vision performs better in recognizing text from images when compared to the other two OCR tools used in this experiment. The AWS rekognition also performs well in many cases, but is limited to reading images only in the formats like *.png, .jpg or .jpeg*. For Tesseract-OCR it has been observed that its performance is relatively poor in comparison to the two online OCR tools considered in this experiment.

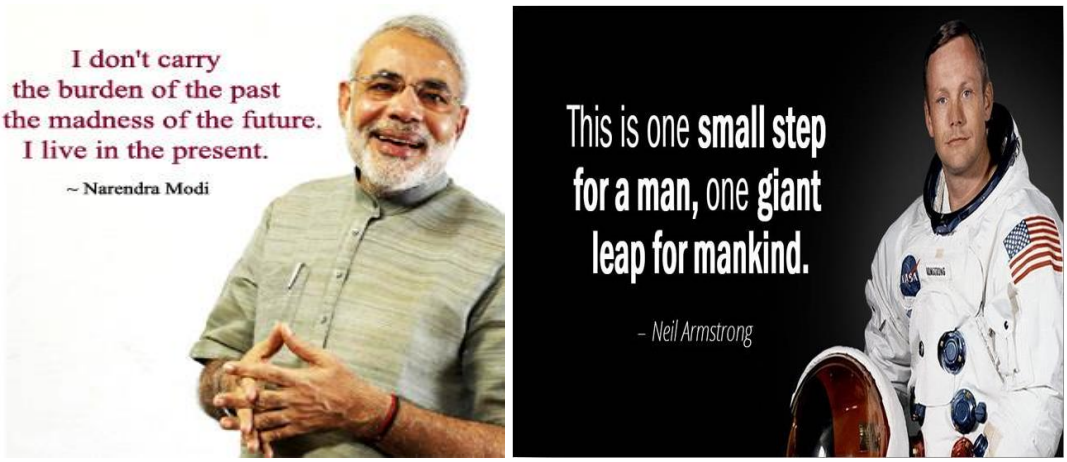
6.2 Verification of Quoted Images

Verification and classification of images having quoted information is essential before its usage. As these images get circulated rapidly in social media, there is a need to validate the source of information. The proposed system aims to validate such images. The results observed after implementation have been described in detail in the subsections. The proposed approach involves the use of OCR tool, Google Cloud Vision, for text detection. A post-processing on the identified text has also been

applied to improve the accuracy of the detected text. The proposed approach uses scripts written in *Python 3* which runs on *Anaconda software (Jupyter Notebook 4.4.0).* The subsections 6.2.1, 6.2.2, and 6.2.3 describe the results obtained, the description of performance metrics and the representation of classification metrics and evaluation results respectively.

6.2.1 Working of the Proposed System on Sample Images

The proposed system follows the below mentioned steps and the results obtained after each step has been discussed below. The flow of the system has been explained with the help of two input images shown below in Figure 6.4.



(a) Quote by Narendra Modi (b) Quote by Neil Armstrong

Figure 6.4: Input images from the dataset [29]

- ***Step 1: Text Detection from an image***

  Using Google Cloud Vision tool

  - *Detected Text from Figure 6.4(a) (Output text 1):*

```
I don't carry
the burden of the past
or the madness of the future.
I live in the present.
~Narendra Modi
```

- *Detected Text from Figure 6.4(b) (Output text 2):*

```
This is one small tep
for a man, one giant
leap for mankind.
Neil Armstrong
72
```

- ***Step 2: Post-process the text extracted***

  Using *language-check* tool.

  - *After Post-Processing Output text 1:*

```
I don't carry
the burden of the past
or the madness of the future.
I live in the present.
~Narendra Mode
```

  - *After Post-Processing Output text 2:*

```
This is one small top
for a man, one giant
leap for mankind.
Neil Armstrong
72
```

- ***Step 3: Extract Author's name***

  Use original Output text for extracting the Author's name instead of post-processed text as is visible from the post-processed result of Output text 1 that the language-check tool has updated the author's name. Therefore, NER and chunking are applied on the original text.

- *Author's name retrieved from Figure 6.4(a):*

  Modi

- *Author's name retrieved for Figure 6.4(b):*

  Neil Armstrong

- ***Step 4: Implement text similarity on the detected text using Google Custom Search and check whether obtained URLs contain terms like misquote, fake, mis-attributed, false-quote, etc. If the terms mentioned above are not present then verify whether extracted domain names are from legitimate list of websites***

  Using Google Custom Search Engine which is a platform provided by Google that allows web developers to feature specialized information in web searches, refine and categorize queries and create customized search engines, based on Google Search.

  - *Extracted URL and domain names from post-processed Output Text 1:*

    i. https://en.wikiquote.org/wiki/Hermann_Hesse
       en.wikiquote.org
    ii. https://www.iep.utm.edu/mean-ear/
       www.iep.utm.edu
    iii. https://en.wikiquote.org/wiki/Wikiquote:Quote_of_the_Day
       en.wikiquote.org
    iv. marvin.cs.uidaho.edu/About/quotes.html
       marvin.cs.uidaho.edu
    v. https://www.ncbi.nlm.nih.gov/books/NBK55415/
       www.ncbi.nlm.nih.gov
    vi. https://www.marxists.org/reference/archive/wilde-oscar/soul-man/
       www.marxists.org
    vii. theconversation.com/guide-to-the-classics-margaret-atwoods-the-handmaids-_tale-75062
       theconversation.com
    viii. https://namica.org/resources/mental-illness/types-mental-illness/
       namica.org
    ix. https://www.powerpoetry.org/related-poems/7793
       www.powerpoetry.org
    x. www.who.int/disabilities/world_report/2011/report.pdf
       www.who.int

Figure 6.4 (a) does not contain any such terms, so there is a need to verify the domain names whether they belong to authentic quote or news site or not.

- *Extracted URL and domain names from post-processed Output Text 2:*

  i. https://www.space.com/17307-neil-armstrong-one-small-step-quote. html
     www.space.com
  ii. https://www.snopes.com/fact-check/one-small-misstep/
     www.snopes.com
  iii. https://www.sbs.com.au/.../armstrong-s-one-small-step-for-man –o ne-giant- misquote-for-mankind

  Extraction of URLs terminates as soon as term misquote was encountered. Moreover, such an image gets classified as *Misquoted* image. As the extracted URLs for Figure 6.4 (b) has the term misquote in it, thus this image is classified as *Misquoted*.

- ***Step 5: Classify the quoted image as Verified, Misquoted or Not Verified.***

  - *Classification of Figure 6.4(a):*

    The status obtained after verification of domain names is *Verified* as can be seen in Figure 6.5.

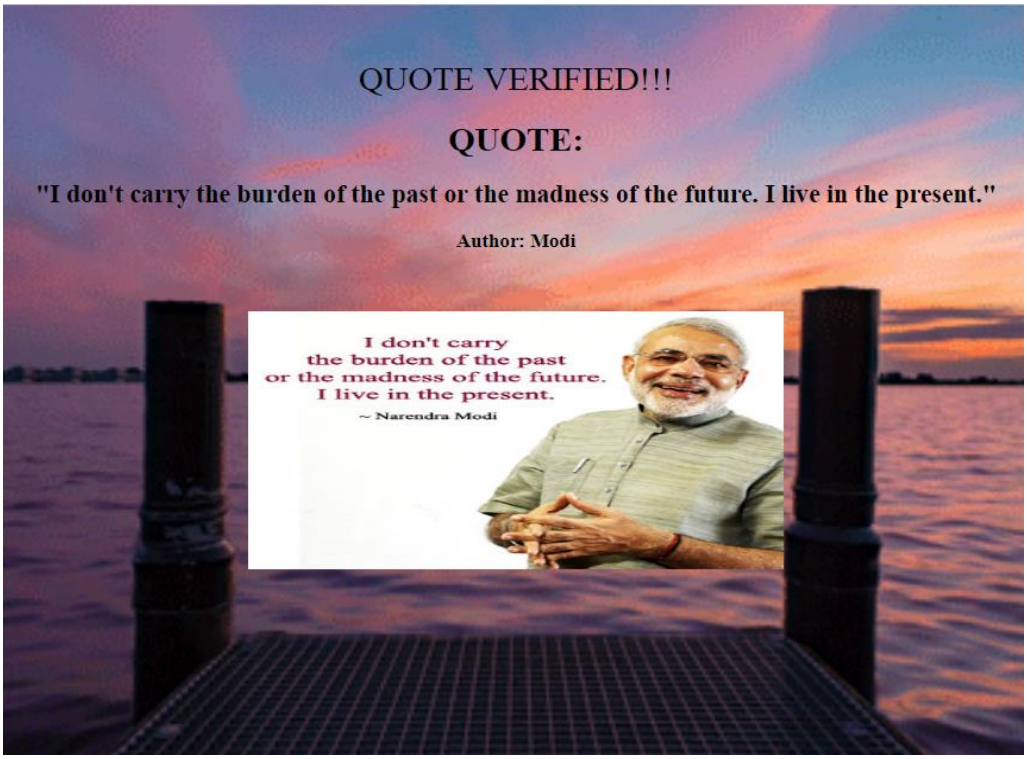


Figure 6.5: Output for the image containing Quote by Narendra Modi [29]

- *Classification of Figure 6.4(b):*

  The status obtained URL extraction process is *Misquoted* as can be seen in Figure 6.6.

Figure 6.6: Output for the image containing Quote by Neil Armstrong [29]

6.2.2 Performance Metrics for Classification

A classifier predicts each sample of the dataset as either correctly identified or incorrectly identified. Taking into consideration 2 classes, *Verified(V)* and *Misquote(M)*, and *V* being the target class, the four possible outcomes of a classification process have been explained below.

- *True Positive (TP):* It indicates correctly predicted event values, *i.e.*, class *V* being classified as class *V*.
- *False Positive (FP):* It indicates incorrectly predicted event values, *i.e.*, class *V* being classified as class *M*.
- *True Negative (TN):* It indicates correctly predicted no-event values, *i.e.*, class *M* being classified as class *M*.
- *False Negative (FN):* It indicates incorrectly predicted no-event values, *i.e.*, class *M* being classified as class *V*.

Accuracy of a classification method is evaluated as the percentage of correctly predicted instances in the entire dataset. The accuracy obtained is between 0 and 1. Below is the formula for calculating accuracy for the classification process.

$$Accuracy\ (A) = (TP + TN) / (TP + FP + TN + FN) \quad (6.1)$$

The observed accuracy for the given confusion matrix comes out to be 96.26%. Evaluating classification accuracy alone at times can be misleading given there is an unequal number of samples in each class. The evaluation metrics discussed below are used for assessing the performance of the suggested approach of classification.

- *Precision:* It is a classification metric that highlights what percentage of instances that has been classified as class *V*, actually belongs to class *V*.

$$Precision\ (P) = TP / (TP + FP) \quad (6.2)$$

- *Recall:* It is a classification metric, also known as *Sensitivity*, which indicates what percentage of instances that actually belong to class *V* was classified by the algorithm as class *V*.

$$Recall\ (R) = TP / (TP + FN) \quad (6.3)$$

- *F1 Score:* Since there are two measures (Precision and Recall), there is a need to have a measurement that represents both of them. F1 score is Harmonic Mean of Precision and Recall. This value is always near to the smaller value of Precision or Recall.

$$F1\ Score = (2 * (8) * (9)) / ((8) + (9)) \quad (6.4)$$

6.2.3 Classification Results of Traditional Models

This section compares the results obtained by using traditional methods (Logistic Regression, Naïves Bayes and SVM) and the proposed approach. Table 6.5 highlights the confusion matrix obtained for the traditional methods. For classification, the quotation dataset as discussed in Section 4.2.2 has been used. The training dataset consists of 7225 labelled quotations and testing dataset consists of an annotated dataset of 3237 labelled quoted images. The observed accuracy for both Logistic Regression and Naïves Bayes comes out to be 84.49%. SVM shows a little improvement with classification accuracy of 85.07%. However, it is visible from the confusion matrix that traditional approach gives unsatisfactory results in case of misquoted images.

Table 6.5. Confusion Matrix of traditional methods

Models		Actual Verified	Actual Misquoted
Logistic Regression	**Predicted Verified**	2734	502
	Predicted Misquoted	0	0
Naïves Bayes	**Predicted Verified**	2734	502
	Predicted Misquoted	0	0
SVM	**Predicted Verified**	2723	472
	Predicted Misquoted	11	30

Table 6.6 represents the evaluation metrics using traditional approach. It can be observed from the Table 6.6 that traditional machine learning models are very poor in recognizing misquotes. F1 score for misquotes is 0% in case of Logistic Regression and Naïves Bayes and F1 score is 1.1% in case of SVM. Therefore, one cannot rely on machine learning techniques for classification of quotes.

Table 6.6. Evaluated Metrics of the traditional method

Metrics	Logistic Regression		Naïves Bayes		SVM	
	Verified	**Misquoted**	**Verified**	**Misquoted**	**Verified**	**Misquoted**
Precision	0.84	0.00	0.84	0.00	0.85	0.73
Recall	1.00	0.00	1.00	0.00	1.00	0.06
F1 Score	0.92	0.00	0.92	0.00	0.92	0.11

6.2.4 Classification Results of Proposed Approach

This section discusses how accurate is the proposed approach in classifying the quoted images. The confusion matrix for the classification of quoted images for the proposed approach has been displayed in Table 6.7. The annotated dataset containing 3237 quoted images of famous personalities has been used to perform the classification process. The observed accuracy for the given confusion matrix comes out to be 96.26% with significant result in both cases.

Table 6.7. Confusion Matrix of the proposed approach

	Actual Verified	**Actual Misquoted**
Predicted Verified	2714	100
Predicted Misquoted	21	402

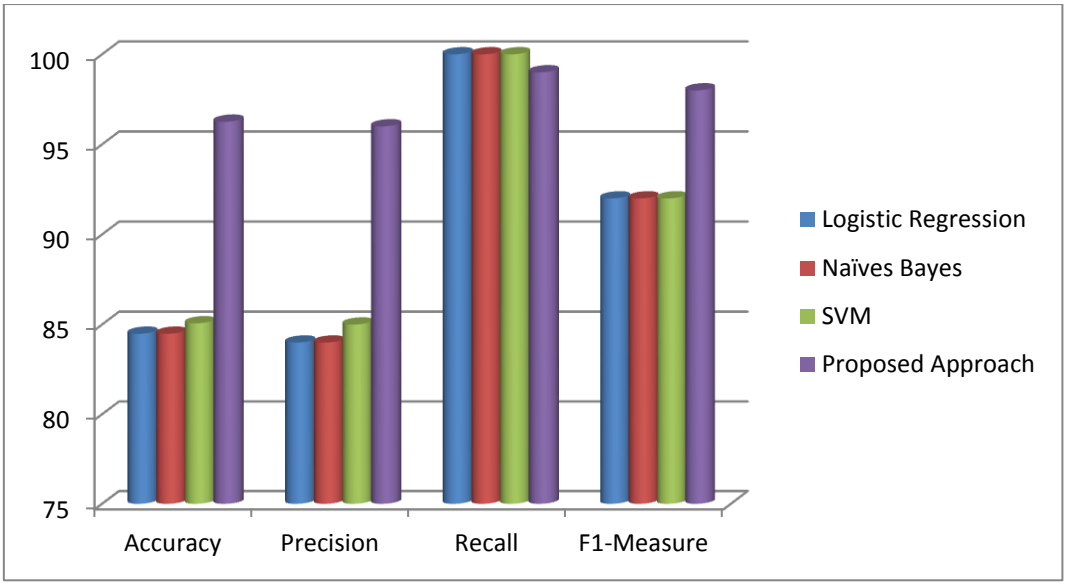
Table 6.8 presents various evaluation metric results obtained for assessing the efficiency of the suggested method of classification of the quoted images. Low recall and high precision, as in the case of Misquoted images, show that a lot of positive samples have been missed, *i.e.*, high FN. Moreover, the ones predicted as positive were indeed positive, *i.e.*, low FP. High recall and high precision, as in the case of Verified images, indicate that the proposed classification model performs very well in identifying images having *Verified* quotes and also gives relatively fair results in case of *Misquotes*.

Table 6.8. Evaluated Metrics of the proposed approach

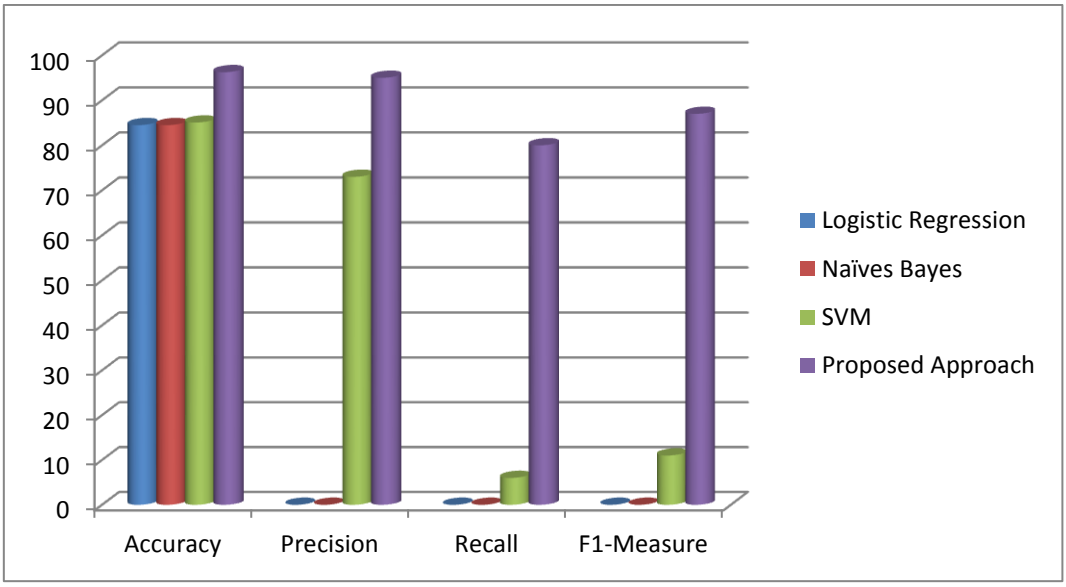
Metrics	**Verified**	**Misquoted**
Precision	0.96	0.95
Recall	0.99	0.80
F1 Score	0.98	0.87

6.3 Comparison Analysis of Classification Models

The confusion matrices and the performance metrics highlight that the proposed system outperforms the traditional machine learning models. Figure 6.7 highlights the superiority of the proposed approach on the traditional models. It is observed that the traditional models showed comparable results in case of Verified quotes. However, in case of Misquotes the performance of the traditional models were significantly poor with Logistic Regression and Naives Bayes giving F1-score as 0%. Also, the proposed approach does not require initial training so can be used in case no training data is available. Also, from the obtained results it can be concluded that this system can be trusted for identifying *Verified* and *Misquoted* images.



(a) Performance analysis for Verified quotes



(b) Performance analysis for Misquotes

Figure 6.7: Performance analysis of Classification [29]

In this chapter, the performance of the OCR tools has been presented and it has been observed that Google Cloud Vision outperforms the OCR tools, Tesseract-OCR and AWS rekognition. Further, this study highlights the binary classification of quoted images into the categories, namely Verified and Misquoted using web-based text similarity and classification. After implementing the proposed system, the classification process achieves an overall accuracy of 96.26%. The overall F1 score was high in both cases indicating correct categorization of the images belonging to the

two categories. The proposed approach can be used for text categorization, fake news detection using images, to verify photo-shopped images, etc. On analysing the results obtained from Google Cloud Visions, it is visible that it has a drawback in case of detecting text from the handwritten images. The conclusion and future work has been discussed in Chapter 7.

CHAPTER 7
CONCLUSION AND FUTURE WORK

7.1 Work Accomplished

After implementation of this system, works accomplished are as follows:

- Extract text from image.
  - Using Tesseract-OCR (open-source tool)
  - Using Google Cloud Vision (needs Google Cloud Access)
  - Using AWS Rekognition (needs Amazon Web Services Access)
- Compare and analyze the performance of the OCR tools.
- Post-Processing of the text retrieved from images.
- Extract Author's name from the retrieved text.
- Fetch URL from Google Custom Search Engine (web-based text similarity).
- Extract domain name from the URLs of similar texts.
- Classification of quoted images as Verified, Not Verified or Misquoted.

7.2 Conclusions

This research focuses on detecting and identifying information from natural scene images. It also proposes a method for improving the accuracy of detected text by using post-processing on the retrieved text for removing wrongly spelled words. The performance of the OCR tools has also been highlighted and it was noticed that each of them performed distinctly on various images. Google Cloud Vision performs well on any image format, like *.jpg, .jpeg, .png, .tiff, .bmp, etc.* and provided mostly relevant results, *i.e.*, recognized only valid text from images. In comparison analysis of the tools, it was noticed that Google Cloud Vision was unable to give any remarkable result in case of handwritten text images or low quality images still; its performance in detecting text from natural scene images has been significant which was required for this experiment. The AWS rekognition worked only on the image formats like *.jpg, .jpeg* and *.png* and even recognized a lot of extraneous marks as characters leading to a low

performance in accurately detecting the text from an image. Further, looking into the case of Tesseract-OCR, which is known to work efficiently in extracting information from scanned documents, gave the weakest performance in comparison to both the online tools. The reason being its inability to recognize the text effectively from natural scene images and text having different fonts and colors, making its evaluated metrics the lowest in comparison of the other two tools.

After demonstrating this approach, there has even been a notable accuracy improvement after implementing post-processing on the retrieved text from the images. The improvement in accuracy is approximately 2% in the case of natural scene images whereas; the improvement is significantly larger, around 8%, in the case of handwritten images. On analyzing the results obtained from all the tools, it is visible that all of them have a common drawback in case of detecting text from the handwritten images. Therefore, there is a need to develop methods to improve information retrieval from handwritten textual images. Since this project deals with natural scene images, therefore Google Cloud Vision has been chosen for extracting text from images.

By implementing the proposed system, the quoted images can be verified and classified as containing Verified and Misquoted images. Also, it was noticed that the proposed system outperforms the traditional machine learning models. The proposed approach was able to achieve accuracy of 96.26% with a F1-score of 92.5%. On the other hand, the traditional models could only achieve an average accuracy of 85% with F1- score of 92% in case of Verified quoted images and only 3.67% in case of Misquoted images. Additionally, it has been observed that the search results obtained by Google Custom Search engine were quite close to Google Search engine and were able to retrieve URL of similar text efficiently. Moreover, classification of images was also achieved with efficient performance using the proposed method.

7.3 Social Benefits

In recent times, digital data has seen a tremendous increase, which has led its users to believe anything found on the internet. After successful implementation, the end users will be able to easily convert digital data found in the format of images in a readable textual format. Moreover, the users will also be able to verify the quotes before its usage. Using misquotes, false quotes or mis-attributed quotes brings down the value of one’s speech or text.

7.4 Future Work Plan

The limitations and drawback of the three OCR tools used for comparison and the approach of quoted image classification performed thereafter have indicated the following areas as recommendation for future work.

- Improve accuracy of text extracted from images having typical fonts, handwritten texts and noisy images.
- Improving author name extraction using NER as in-built library works better on American and British names.
- Enhance the author name extraction using Face recognition techniques alongwith the NER extraction.
- Using proposed approach for classifying fake news circulated through online platform or photoshopped images.
- Enhance the domain authentication procedure using features like trustworthiness, fraud cues; distinguish the fake site from legitimate one, *etc.*
- The database of authentic quotation sites can be extended and improved.

REFERENCES

[1] E. Gur, Z. Zelavsky. “Retrieval of Rashi Semi-cursive Handwriting via Fuzzy Logic”. International Conference on Frontiers in Handwriting Recognition, Bari, 2012, pp. 354-359.

[2] S. Dutta, N. Sankaran, K. P. Sankar, C. V. Jawahar. “Robust Recognition of Degraded Documents Using Character N-Grams”. 10th IAPR International Workshop on Document Analysis Systems, Gold Cost, QLD, 2012, pp. 130-134.

[3] A. AlSalman, A. El-Zaart, S. Al-Salman, A. Gumaei. “A novel approach for Braille images segmentation”. International Conference on Multimedia Computing and Systems (pp. 190-195). IEEE, 2012.

[4] S. Du, M. Ibrahim, M. Shehata, W. Badawy. “Automatic License Plate Recognition (ALPR): A State-of-the-Art Review”. IEEE Transactions on Circuits and Systems for Video Technology, 2012, vol. 23, no. 2, pp. 311-325.

[5] J. Yang, K. Wang, J. Li, J. Jiao, J. Xu. “A fast adaptive binarization method for complex scene images”. 19th IEEE International Conference on Image Processing, Orlando, FL, 2012, pp. 1889-1892.

[6] Y. Bassil, M. Alwani. “Ocr post-processing error correction algorithm using google online spelling suggestion”. Journal of Emerging Trends in Computing and Information Sciences, ISSN 2079-8407, 2012, Vol. 3, No. 1.

[7] K. Ntirogiannis, B. Gatos, I. Pratikakis. “Performance Evaluation Methodology for Historical Document Image Binarization”. IEEE Transactions on Image Processing, 2013, vol. 22, no. 2, pp. 595-609.

[8] P. M. Manwatkar, S. H. Yadav. “Text recognition from images”. International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, 2015, pp. 1-6.

[9] V. Rajan, S. Raj. “Text detection and character extraction in natural scene images using fractional poisson model”. International Conference on

Computing Methodologies and Communication (ICCMC), Erode, 2017, pp. 1136-1141.

[10] R. Kushol, I. Ahsan, M. N. Raihan. "An Android-Based Useful Text Extraction Framework Using Image and Natural Language Processing". International Journal of Computer Theory and Engineering, 2018, Vol. 10, No. 3.

[11] W. H. Gomaa, A. A. Fahmy. "A survey of text similarity approaches". International Journal of Computer Applications, 2013, 68(13), 13-18.

[12] N. Pradhan, M. Gyanchandani, R. Wadhvani. "A Review on Text Similarity Technique used in IR and its Application". International Journal of Computer Applications, 2015, 120(9).

[13] C. Samarinas, G. Tsoumakas. "WAMBy: An information retrieval approach to web-based question answering". Proceedings of the 10th Hellenic Conference on Artificial Intelligence (p. 40). 2018. ACM.

[14] S. Kowser, S. Rahman, D. A. Shojol, M. Kabir. "Study on Text Similarity Measure Algorithms For English Language". Doctoral dissertation, United International University, 2019.

[15] A. Mukherjee, V. Venkataraman, B. Liu, N. Glance. "Fake review detection: Classification and analysis of real and pseudo reviews". UIC-CS-03-2013. Technical Report, 2013.

[16] M. Iyyer, V. Manjunatha, J. Boyd-Graber, H. Daumé III. "Deep unordered composition rivals syntactic methods for text classification". Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015, (Volume 1: Long Papers) (Vol. 1, pp. 1681-1691).

[17] A. Tripathy, A. Agrawal, S. K. Rath. "Classification of Sentimental Reviews Using Machine Learning Techniques". Procedia Computer Science, 57, 2015, 821-829.

[18] L. Arras, F. Horn, G. Montavon, K. R. Müller, W. Samek. ""What is relevant in a text document?": An interpretable machine learning approach". PloS one, 2017, 12(8), e0181142.

[19] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, R. Mihalcea. “Automatic detection of fake news”. arXiv preprint arXiv:1708.07104, 2017.

[20] IIIT 5K-word dataset. Available on: *http://cvit.iiit.ac.in/projects/SceneText Understanding /IIIT5K.html*

[21] KAIST Scene Text Dataset. Available on: *http://www.iapr-tc11.org/ mediawiki/index.php/ KAIST_Scene_Text_Database*

[22] Handwritten word dataset. Available on: *https://www.kaggle.com/ nabeel965/handwritten-words-dataset*

[23] Google Cloud Vision API. Available on*:* *https://cloud.google.com/ vision/docs/libraries*

[24] AWS rekognition. Available on: *https://docs.aws.amazon.com/rekognition /latest/dg/text-detection.html*

[25] B. S. Kumar. “Image denoising based on non-local means filter and its method noise thresholding”. Signal, image and video processing, 7(6), 2013, 1211-1227.

[26] Y. Skalban, L. Specia, R. Mitkov. “Automatic question generation in multimedia-based learning”. Proceedings of COLING 2012: Posters: 1151-1160.

[27] W. Huang, Y. Qiao, X. Tang. “Robust scene text detection with convolution neural network induced mser trees”. European Conference on Computer Vision (pp. 497-511). 2014. Springer, Cham.

[28] Z. Tian, W. Huang, T. He, P. He, Y. Qiao. “Detecting text in natural image with connectionist text proposal network”. European conference on computer vision (pp. 56-72). 2016. Springer, Cham.

[29] *https://drive.google.com/open?id=1O9aNCEDowiFpZ6m8ID6mFq5oS_TipFlU*

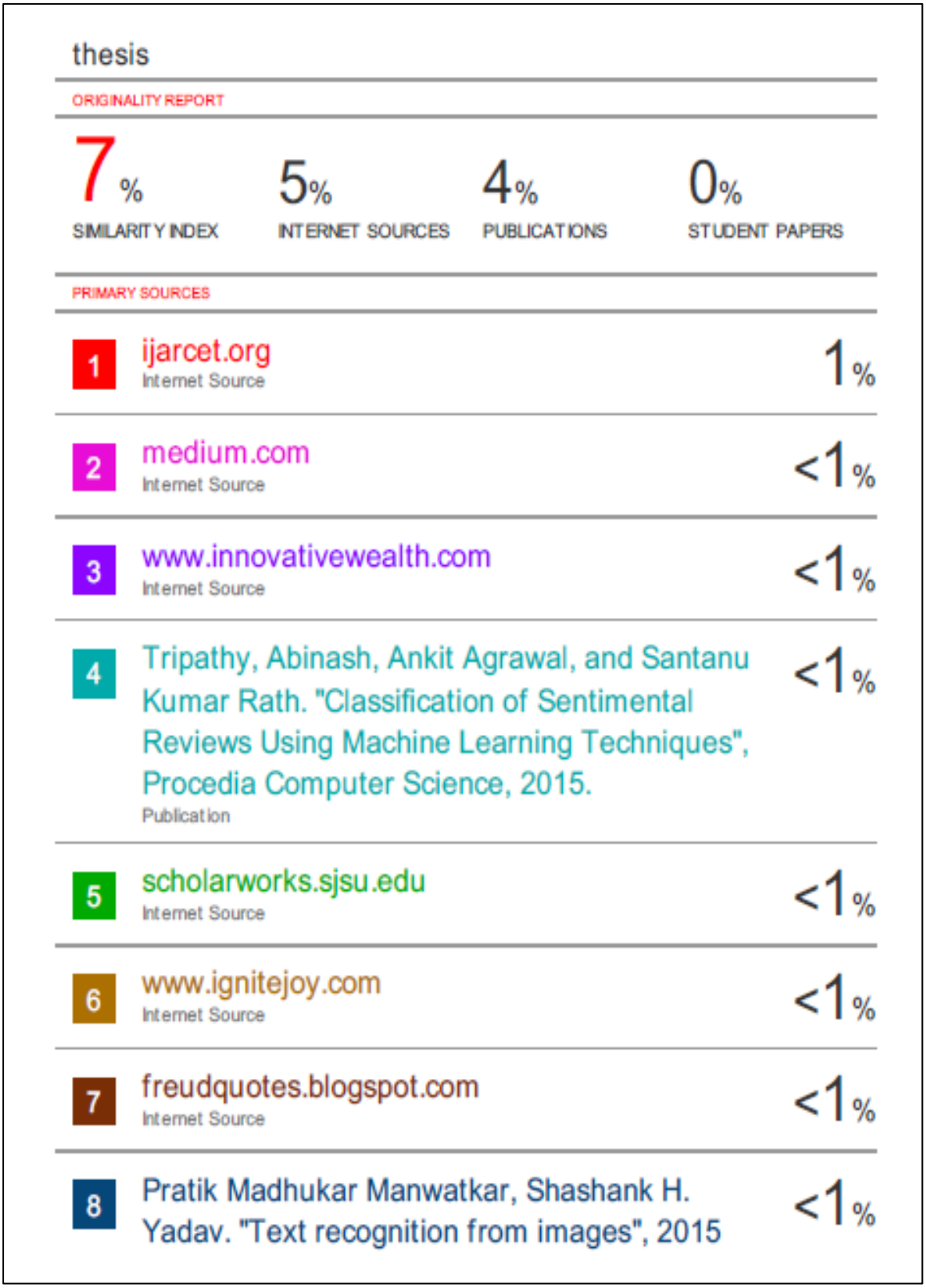
[30] *https://aws.amazon.com/blogs/machine-learning/amazon-rekognition-announces-real-time-face-recognition-support-for-recognition-of-text-in-image-and-improved-face-detection/*

LIST OF PUBLICATIONS

[1] S. Banerjee, P. Kumar, S. Kaur. "Performance Analysis of Optical Character Recognition Tools in Natural Scene Text detection using OCR Post-processing". International Conference on Advanced Data Analysis, Business Analytics and Intelligence (ICADABAI), 2019. [Accepted]

[2] S. Banerjee, P. Kumar, S. Kaur. "Quote Examiner: Verifying Quoted Images Using Web-based Text Similarity". Computer Society of India (CSI) Transactions on ICT, 2019. [Communicated]

PLAGERISM REPORT



thesis

ORIGINALITY REPORT

7%	5%	4%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ijarcet.org Internet Source	1%
2	medium.com Internet Source	<1%
3	www.innovativewealth.com Internet Source	<1%
4	Tripathy, Abinash, Ankit Agrawal, and Santanu Kumar Rath. "Classification of Sentimental Reviews Using Machine Learning Techniques", Procedia Computer Science, 2015. Publication	<1%
5	scholarworks.sjsu.edu Internet Source	<1%
6	www.ignitejoy.com Internet Source	<1%
7	freudquotes.blogspot.com Internet Source	<1%
8	Pratik Madhukar Manwatkar, Shashank H. Yadav. "Text recognition from images", 2015	<1%