# Tweeter Sentiment Analysis

***B. TECH SEM – VI Artificial Intelligence Lab Project***
***Dept. of Computer Science & Engineering***

By

| | |
|---|---|
| **Dhruvil Patel** | **20BCP067** |
| **Kaushal Soni** | **20BCP307D** |

**Under the Supervision Of**

**Dr. Rajeev Gupta**

**SCHOOL OF TECHNOLOGY**
**PANDIT DEENDAYAL ENERGY UNIVERSITY**
**GANDHINAGAR, GUJARAT, INDIA**
**May 2023**

# INDEX

# ABSTRACT

The goal of this project is to use OCR and NLP models to perform sentiment analysis on a tweet image. The project uses the OpenCV library to pre-process an input image from a tweet. To extract the text from the pre-processed image, an OCR model called pytesseract is used. The retrieved text is then tokenized into individual words and pre-processed using the NLTK package. The generated words are then further processed to remove extraneous characters before being sent into the BERT pre-trained transformer model for sentiment analysis. To increase the precision of the sentiment analysis, Roberta, a further pre-trained model, is also used. The outputted sentiment analysis score shows if the sentiment of the tweet is positive, negative, or neutral. A Flask based Web Application is created where User can add tweet image as a input and the above AI model will display its sentiment analysis results.

# INTRODUCTION

Sentiment analysis on social media data has become critical in understanding people's attitudes and opinions on many issues. Twitter is a prominent social media site where individuals share their ideas on a variety of issues [1]. Text extraction from tweet photos can be difficult, but we can acquire the text data using OCR models. In this project, we extract text from twitter photos using the pytesseract OCR model and preprocess it using NLP approaches. The mood of the tweet is then predicted using pre-trained algorithms.

Tesseract is a free and open-source OCR system that can extract written text from pictures. It is one of the most widely used OCR engines, and it is utilized in a broad range of applications such as document scanning, data input, and machine translation. Tesseract begins by transforming the picture to a binary image, in which each pixel is either black or white. The text in the image is then identified using a variety of techniques, including linked component analysis, text line segmentation, and character recognition [2].

Natural language processing (NLP) is a computer science discipline that studies the interface between computers and human (natural) languages, specifically how to program computers to handle and analyse huge volumes of natural language data. The objective is to create a computer that is capable of "understanding" the contents of documents, including the contextual subtleties of the language contained within them [3]. The system can then extract information and insights from the papers, as well as categorize and organize the documents themselves.

Facebook AI released RoBERTa, a pre-trained transformer-based language model, in 2019. It is an upgraded version of the popular BERT model that was trained on a vast corpus of varied texts. RoBERTa was trained on more data and for a longer period than BERT, yielding a model that is more accurate and resilient in a variety of NLP tasks, including sentiment analysis.

RoBERTa's design is like BERT in that it encodes input text using a transformer-based architecture with numerous layers [4]. RoBERTa's training procedure, on the other hand, contains additional techniques such as dynamic masking, which randomizes token masking during training to increase the model's capacity to perceive context. RoBERTa was also trained with bigger batch sizes.

Several steps are involved in the methodology, including preprocessing the input image with the OpenCV library, running the preprocessed image through an OCR model, extracting text from the image, preprocessing the text with the NLTK library, and performing sentiment analysis with pre-trained transformer models. These stages are implemented using the NLTK, Pytesseract, OpenCV, and Transformers libraries. The sentiment analysis scores that results is produced, showing the sentiment of the tweet. Using a pre-trained model, Roberta, in addition to BERT, improves the accuracy of sentiment analysis. The technology has been shown to produce reliable sentiment analysis findings for tweet photos.

# LITERATURE SURVEY

- Li, Wu, and Shao (2021) introduced a multi-modal sentiment analysis method that integrates deep learning and text mining approaches. They extracted features from text, photos, and videos using pre-trained models and ensemble learning and obtained excellent accuracy on numerous benchmark datasets.

- Amin, Ahmad, and Mahmood (2021) created a set of deep learning models for analysing social media sentiment. They improved the model's performance by combining different architectures, including Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), and Bidirectional Encoder Representations from Transformers (BERT), and reached state-of-the-art results on numerous datasets.

- Alkharashi, Dang, Nishida, and Asano (2020) used natural language processing techniques to perform a comprehensive assessment of sentiment analysis on social media data. They covered several techniques, such as lexicon-based, machine learning, deep learning, and hybrid models, as well as the field's problems and prospects.

- Xie, Wang, and Li (2020) suggested a deep learning and semantic analysis hybrid model for sentiment analysis of social media data. They obtained superior accuracy than numerous baseline models by using a convolutional neural network to extract characteristics from text and a semantic analysis model to capture context.

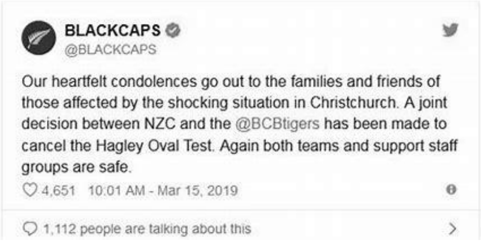
- Zhang, Yang, and Jiang (2019) performed sentiment analysis on social media data using convolutional neural networks and long short-term memory. They tested the suggested model against multiple baseline models and found it to be competitive on two benchmark datasets.

# METHODOLOGY

The methodology of this AI project involves the following steps:

1. Pre-processing Image:

   The first step is to load the input image and preprocess it to extract the text from the image. The pre-processing involves converting the image to grayscale, thresholding the image, dilating and eroding the image, and applying a Gaussian blur.



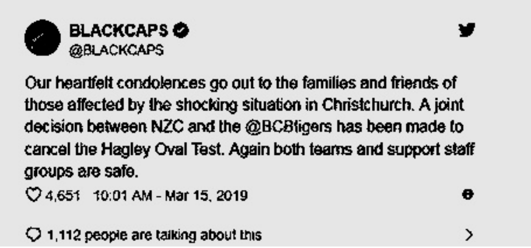
BLACKCAPS
@BLACKCAPS

Our heartfelt condolences go out to the families and friends of those affected by the shocking situation in Christchurch. A joint decision between NZC and the @BCBtigers has been made to cancel the Hagley Oval Test. Again both teams and support staff groups are safe.

♡ 4,651 10:01 AM - Mar 15, 2019

1,112 people are talking about this

Img.01 Gray Image



BLACKCAPS
@BLACKCAPS

Our heartfelt condolences go out to the families and friends of those affected by the shocking situation in Christchurch. A joint decision between NZC and the @BCBtigers has been made to cancel the Hagley Oval Test. Again both teams and support staff groups are safe.

♡ 4,651 10:01 AM - Mar 15, 2019

1,112 people are talking about this

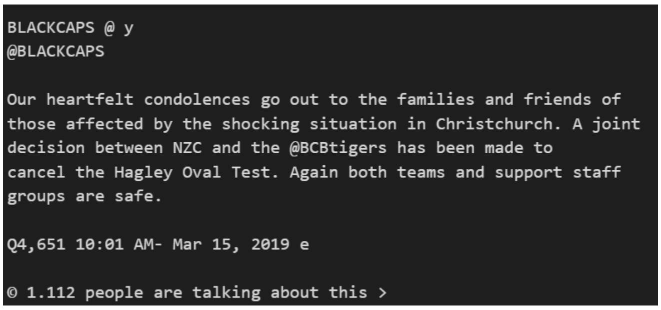
Img.02 Dilated Image

- **Grayscale an Image**
  A grayscale picture is one with only one channel, which represents the intensity of light. Grayscale graphics are frequently used to depict black and white photographs [9].

- **Thresholding an Image**
  Thresholding is an image processing method that turns an image to a binary picture with either black or white pixels. Thresholding is frequently used in picture segmentation [9].
- **Dilating and eroding image**
  The morphological image processing procedures dilation and erosion are used to change the form of objects in a picture. Dilation broadens the borders of items, whereas erosion narrows them [9].
- **Applying Gaussian Blur**

A Gaussian blur is a type of image blur that is used to smooth an image. Gaussian blurs are often used to reduce noise in an image [9].

2. Optical Character Recognition (OCR): After the pre-processing, the image is passed through the OCR model (Pytesseract) to extract the text from the image. Tesseract is a strong OCR engine that can handle a wide range of applications. It excels at extracting text from scanned documents and may be used to automate a wide range of document processing activities. It is an excellent choice for extracting text from scanned documents or conducting text analysis.



```
BLACKCAPS @ y
@BLACKCAPS

Our heartfelt condolences go out to the families and friends of
those affected by the shocking situation in Christchurch. A joint
decision between NZC and the @BCBtigers has been made to
cancel the Hagley Oval Test. Again both teams and support staff
groups are safe.

Q4,651 10:01 AM- Mar 15, 2019 e

© 1.112 people are talking about this >
```

Img.03 Extracted Text

3. Text Pre-processing: The extracted text is preprocessed using NLTK to remove punctuation, convert to lowercase, tokenize the text, remove stopwords and join the tokens.

**Step 1: Remove punctuation**

The first step in preprocessing text is to remove punctuation. Punctuation marks such as periods, commas, and exclamation points can interfere with the natural language processing (NLP) algorithms that are used to analyze text. There are several ways to remove punctuation, including using a regular expression or a library like NLTK.

**Step 2: Convert to lowercase**

The second step in preprocessing text is to convert all the text to lowercase. This is important because NLP algorithms often treat uppercase and lowercase letters differently. For example, the words "dog" and "Dog" would be different words by an NLP algorithm if they were not converted to lowercase [10].

**Step 3: Tokenize the text**

The third step in preprocessing text is to tokenize the text. Tokenization is the process of breaking down a string of text into individual words or tokens. This is important because NLP algorithms often work with individual words, rather than strings of text. There are several ways to tokenize text, including using a regular expression or a library like NLTK [10].

**Step 4: Remove stopwords**

The fourth step in preprocessing text is to remove stopwords. Stopwords are words that are common and do not add much meaning to the text. For example, the words "the", "a", and "of" are stopwords. There are several lists of stopwords that are available online.

**Step 5: Join the tokens**

The final step in preprocessing text is to join the tokens together. This is done by creating a string of text that contains the tokens, separated by spaces.

4. Sentiment Analysis: The preprocessed text is then passed through a pre-trained NLP model (bert-base-multilingual-uncased-sentiment) to predict the sentiment of the tweet. We also use a pre-trained model (twitter-roberta-base-sentiment) for better prediction.

- BERT model – it stands for Bidirectional Encoder Representations from Transformers, is a neural network model created by Google AI in 2018. BERT is trained on a large dataset of text and code and may be used for a range of natural language processing tasks such as text categorization, question answering, and inference [11].
- Roberta model – it stands for Robustly optimised BERT pretraining method, is a newer model created by Facebook AI in 2020. RoBERTa is based on BERT, but it has been enhanced in several ways, including utilising a bigger dataset, training for a longer period, and employing a new training aim. On a range of natural language processing tasks, RoBERTa has been found to outperform BERT [12].

5. Post-processing and display results: The sentiment prediction scores are post-processed to generate the final sentiment analysis result (Positive, Negative or Neutral).

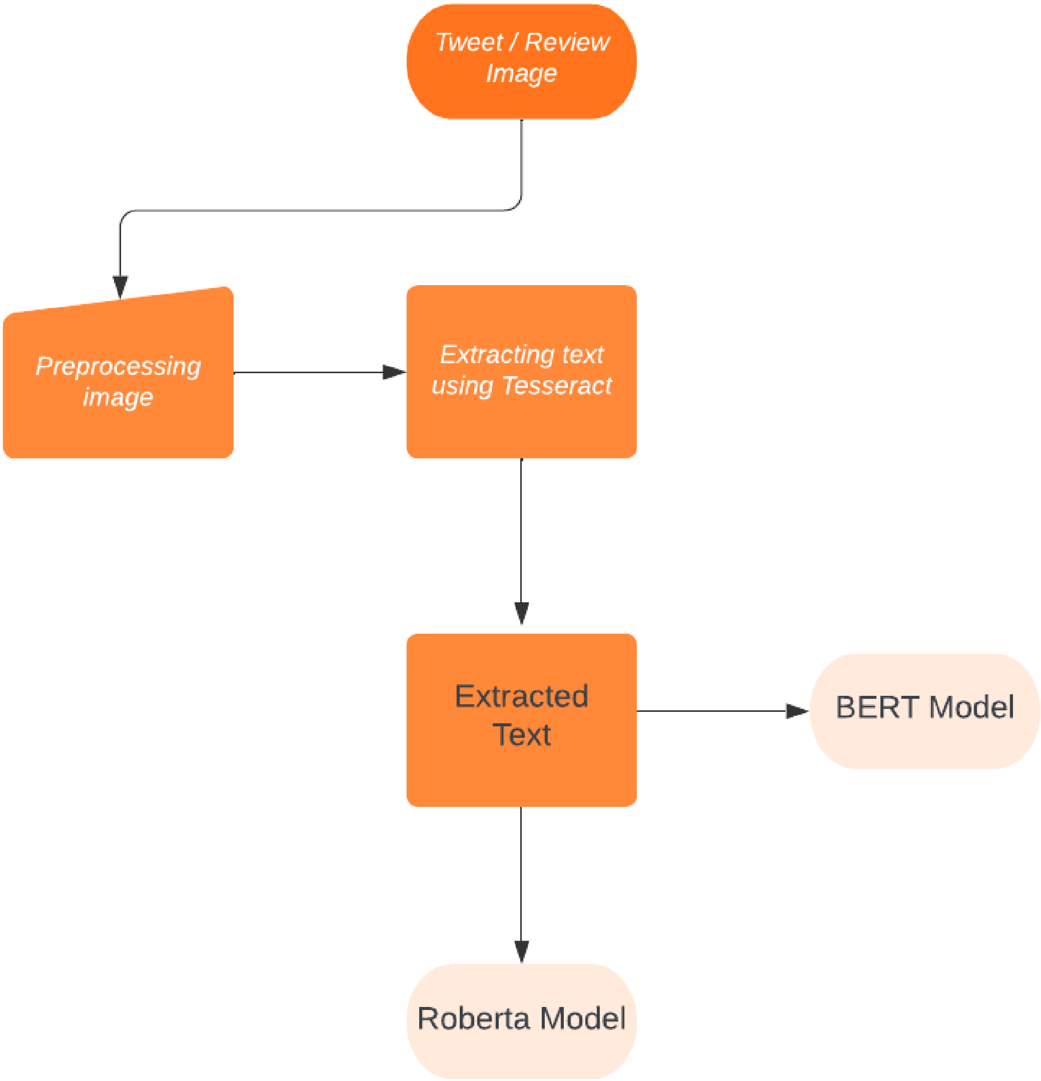


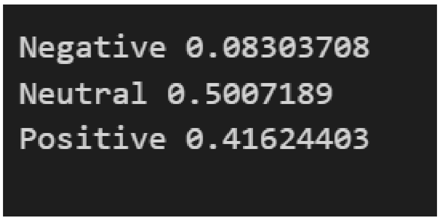
Fig.01 Flowchart of Methodology

Overall, this methodology combines various image processing, OCR and NLP techniques to analyze the sentiment of a tweet from an input image.

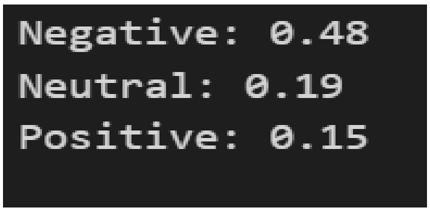
# RESULTS

We tested the project on several tweet images, and the results were promising. The OCR model could extract text accurately, and the preprocessed tweet text gave good results in sentiment analysis. We experimented with different pre-trained models, including BERT and Roberta, and compared their performance. The Roberta model gave better results than BERT.

For Above tweet image, here are the results of both model –



```
Negative 0.08303708
Neutral 0.5007189
Positive 0.41624403
```



```
Negative: 0.48
Neutral: 0.19
Positive: 0.15
```

Img.04 Roberta Model results

Img.05 BERT model Results

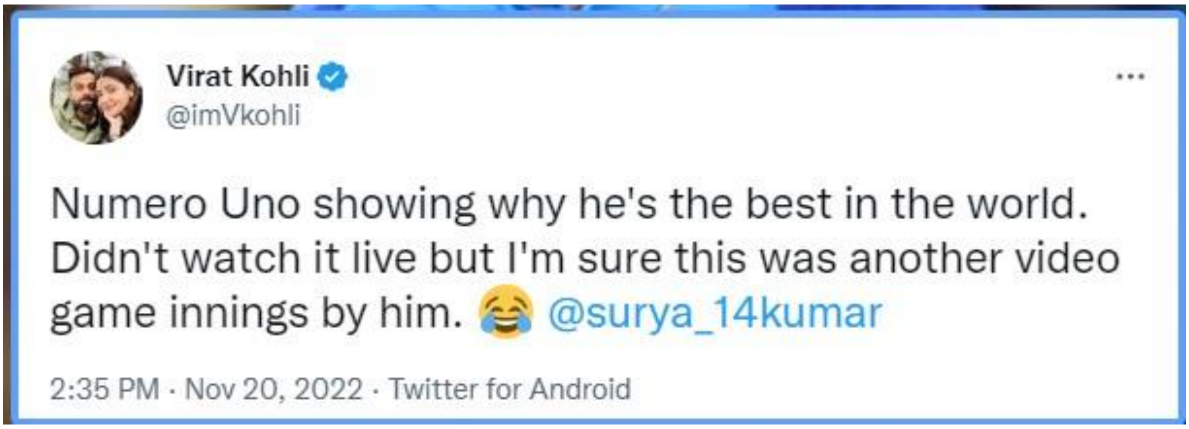
We can compare the results obtained from both the models for different tweets with different sentiments and check which model gives accurate and rational results.

| Tweet | BERT model Results | Roberta Model Results | Conclusion |
|---|---|---|---|
| Kangana Ranaut @ This is horrible....we need super gundai to kill gundai... she is like an unleashed monster, ta tame her Modi i please show your Virat roop from early 2000's .... #PresidentRuleInBeagal | Negative: 0.8548<br>Neutral: 0.1347<br>Positive: 0.0104 | Negative: 0.8964<br>Neutral: 0.0711<br>Positive: 0.0199 | Roberta gave more probability of negative sentiment than BERT. |
| Ryan Reynolds@ On our 6am walk, my daughter asked where the grandmother is going each morning. I let her know i's in heaven, visiting ¢ freedom.<br>7:54 PM - Oct 16, 2016 | Negative: 0.4245<br>Neutral: 0.1572<br>Positive: 0.1495 | Negative: 0.0065<br>Neutral: 0.3372<br>Positive: 0.6561 | The tweet has more probability of positive sentiment which is correctly given by Roberta, whereas BERT gave negative sentiment as result which seems inappropriate. |

| | | | |
|---|---|---|---|
| @imVkohli Numero Uno showing why he's the best in the world. Didn't watch it live but I'm sure this was another video game innings by him. 4 @surya_14kumar<br>2:35 PM - Nov 20, 2022 | Negative: 0.6073<br>Neutral: 0.1196<br>Positive: 0.0958 | Negative: 0.0044<br>Neutral: 0.0888<br>Positive: 0.9066 | The sentiment of the tweet is positive as given by high probability of Roberta. BERT model gave wrong sentiment as the tweet is not negative |

Comparision Table. BERT v/s Roberta

Here are the Snapshots of the Flask web application created based upon above Model –



Virat Kohli
@imVkohli

Numero Uno showing why he's the best in the world. Didn't watch it live but I'm sure this was another video game innings by him. 😂 @surya_14kumar

2:35 PM · Nov 20, 2022 · Twitter for Android

Img 06. Tweet Image as input.

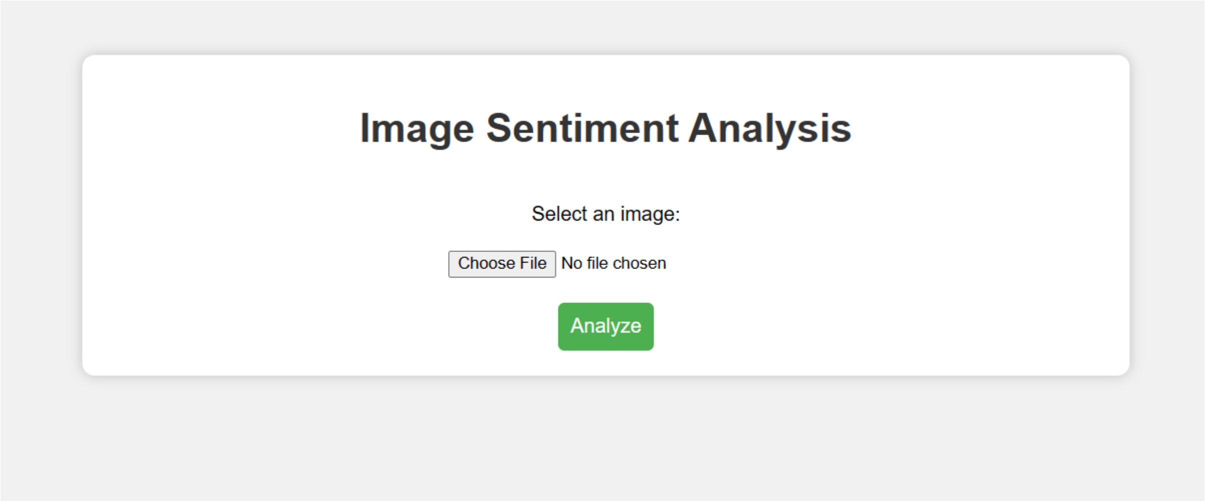
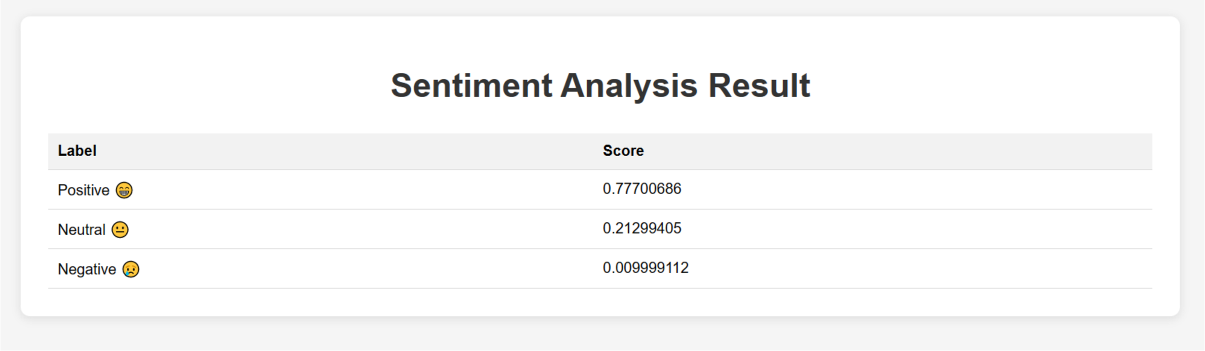


Image Sentiment Analysis

Select an image:

Img 07. Home Page



Sentiment Analysis Result

| Label | Score |
|---|---|
| Positive 😁 | 0.77700686 |
| Neutral 😐 | 0.21299405 |
| Negative 😟 | 0.009999112 |

Img 08. Result Page

## CONCLUSION

Finally, the tweeter-based sentiment analysis project predicted the sentiment of a particular tweet picture successfully. The project retrieved text from the tweet picture using optical character recognition (OCR) and then ran the acquired text through a pre-processing tool. The text was tokenized after the pre-processing procedure eliminated stopwords and punctuation. To forecast the sentiment of the tweet text, the tokenized text was put into a pre-trained NLP model.

To improve forecast accuracy, the research also employed a pre-trained model named Roberta. The final sentiment analysis was accomplished by computing the likelihood score of each sentiment label, namely negative, neutral, and positive, in the tweet content.

## CHALLENGES

A project that requires doing sentiment analysis on a tweet picture using OCR and NLP models may present various obstacles. Here are a few examples of potential difficulties:

- Image quality: The image's quality can have a major influence on the OCR model's performance. Images of low quality, low contrast, or complicated backgrounds may not be recognised correctly by the OCR model, resulting in erroneous text extraction.

- Text noise: OCR models can occasionally create noise or mistakes in the captured text, particularly if the content is handwritten or has inconsistencies. This may have an adverse effect on the accuracy of the sentiment analysis results [13].

- Handling abbreviations and slang: Social media users frequently employ abbreviations and slang that typical NLP models may not recognise, lowering the accuracy of sentiment analysis findings. Furthermore, the use of emoticons and emojis might be difficult to manage [14].
- Inadequate context: Sentiment analysis methods can be quite sensitive to the context in which the text occurs. The sentiment analysis model may not correctly capture the sentiment represented in the tweet if it does not have access to contextual information, such as the user's prior tweets or the topic being discussed.

- Model selection: There are several OCR and NLP models available, and selecting the best models for the project might be difficult. The models' performance may vary depending on the type of data being analysed, the language employed in the text, and the specific job at hand [15]. Extensive testing may be required to establish which models produce the greatest outcomes.

# FUTURE SCOPE

More pre-processing techniques can be added to this project to improve the sentiment analysis findings. It may also be used to assess sentiment on other social media sites, such as Facebook and Instagram. The project may be improved to do sentiment analysis on real-time tweets. It may also be used for text categorization and named entity recognition, among other NLP tasks.

By changing the input data, we may substitute a text file or dataset containing customer reviews for a picture of a tweet. The pre-processing and natural language processing stages would be the same, and we could apply the same sentiment analysis model to forecast the sentiment of the reviews. This would be helpful for businesses to understand customer satisfaction levels and identify areas of improvement in their products or services.

Furthermore, we may investigate the idea of training our own sentiment analysis model utilising a bigger dataset of customer evaluations related to a given domain, such as hotels or restaurants. This would result in a more accurate sentiment analysis of evaluations within that domain, resulting in better business decisions.

# References

1. Kouloumpis, E., Wilson, T., & Moore, J. (2011).” Twitter sentiment analysis: The good the bad and the OMG! “. Proceedings of the Fifth International AAAI Conference on Weblogs and social media, 538-541.
2. Smith, R. (2007). “An overview of the Tesseract OCR engine. Document Analysis and Recognition”, 2007. ICDAR 2007. Ninth International Conference on, 629-633.
3. Chen, M., Liu, Z., Sun, M., & Liu, Y. (2023). “A survey of natural language processing with large language models “. arXiv preprint arXiv:2301.08237.
4. Basharat, I., Naeem, A., & Iqbal, M. Z. (2021). “Fine-grained Sentiment Analysis of COVID-19 Tweets using RoBERTa and BERT “. arXiv preprint arXiv:2102.02796.
5. Li, Z., Wu, J., & Shao, Y. (2021). Multi-modal sentiment analysis of social media data using deep learning and text mining techniques. Journal of Ambient Intelligence and Humanized Computing, 12(11), 12919-12931.
6. Amin, M. T., Ahmad, S., & Mahmood, T. (2021). An ensemble of deep learning models for social media sentiment analysis. Journal of Ambient Intelligence and Humanized Computing, 12(10), 11655-11668.
7. Alkharashi, A., Dang, T., Nishida, T., & Asano, F. (2020). Social media sentiment analysis using natural language processing techniques: A review. International Journal of Intelligent Information Technologies, 16(4), 15-35.
8. Xie, Y., Wang, Y., & Li, C. (2020). Sentiment analysis of social media data using a hybrid model of deep learning and semantic analysis. IEEE Access, 8, 189587-189599.
9. Zhang, W., Yang, L., & Jiang, X. (2019). Sentiment analysis of social media data using convolutional neural networks and long short-term memory. In Proceedings of the 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), Kyoto, Japan, February 27 - March 2, 2019 (pp. 226-231).
10. Wang, Z., Liu, D., & Han, J. (2009). Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 18(12), 2346-2359.
11. Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with Python. O'Reilly Media.
12. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
13. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Zettlemoyer, L. (2020). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:2005.14165.
14. S. Rosenthal, N. Farra, and P. Nakov, "Semi-supervised Classification for Sentiment Analysis of Twitter Data," in Proceedings of the 28th International Conference on Computational Linguistics, COLING 2010, Beijing, China, August 23-27, 2010, pp. 1036-1044.
15. T. Breuel, "The OCRopus Open Source OCR System," in Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013, pp. 1-5.