iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** V **Month of publication:** May 2023

DOI: https://doi.org/10.22214/ijraset.2023.52889

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

Convolutional Neural Networks for Text Classification: A Comprehensive Analysis

Prof. Sachin Sambhaji Patil[1], Anthon Rodrigues[2], Rahul Telangi[3], Vishwajeet Chavan[4]
[1]*Assistant Professor, Department Of Computer Engineering, ZCOER, Pune*
[2, 3, 4] *BE Students, Zeal College of Engineering and Research, Pune, Maharashtra, India*

Abstract: ***This research paper explores the integration of Convolutional Neural Networks (CNNs) with the VADER (Valence Aware Dictionary and sEntiment Reasoner) sentiment analysis tool for text classification tasks. CNNs have shown promising results in text classification, while VADER is a well-established lexicon and rule-based sentiment analysis tool. By combining the strengths of both approaches, we aim to enhance the accuracy and effectiveness of text classification models. The proposed approach leverages the local context capture capabilities of CNNs and the sentiment analysis capabilities of VADER to classify text into predefined categories. We evaluate the performance of the CNN with VADER model on benchmark datasets, comparing it with other state-of-the-art text classification models. The results demonstrate that the integration of CNNs with VADER significantly improves classification accuracy and provides a more nuanced understanding of sentiment in textual data. This research contributes to the field of text classification by highlighting the benefits of combining deep learning models with sentiment analysis tools for more accurate and informative classification.***

Keywords: Text classification, Convolutional Neural Networks, VADER sentiment analysis, Deep learning, Sentiment analysis, Classification accuracy.

I. INTRODUCTION

Text classification, the task of assigning predefined categories or labels to textual data, is a fundamental problem in natural language processing (NLP). It plays a crucial role in various applications, including sentiment analysis, topic classification, spam detection, and document categorization. Convolutional Neural Networks (CNNs) have gained significant attention in recent years as powerful deep learning models for text classification tasks due to their ability to capture local patterns and hierarchical representations within the text. In this research paper, we focus on applying CNNs for text classification specifically in the context of Twitter data. Twitter has become a prominent platform for real-time information sharing, making it a valuable source for analyzing public sentiment, tracking trends, and monitoring social discussions. To explore the effectiveness of CNNs for Twitter text classification, we utilize a large dataset consisting of 1.6 million tweets.

In addition to employing CNNs, we incorporate the VADER (Valence Aware Dictionary and sEntiment Reasoner) sentiment analysis tool into our framework. VADER is specifically designed for social media texts, including Twitter data, and provides a lexicon-based approach to sentiment analysis by considering contextual valence and sentiment intensities. By integrating VADER with CNNs, we aim to enhance the performance of sentiment classification on the Twitter dataset.

Our objectives in this research paper are twofold: first, to investigate the effectiveness of CNNs for text classification on a large-scale Twitter dataset, and second, to evaluate the impact of incorporating VADER into the CNN framework for sentiment analysis. We conduct extensive experiments to compare the performance of different CNN architectures and variations, as well as the performance of the CNN-VADER hybrid model, with respect to accuracy, precision, recall, and F1 score.

The findings of this research have implications for both academia and industry. Understanding the capabilities and limitations of CNNs for text classification on Twitter data can contribute to the development of more accurate sentiment analysis systems and better understanding of social media dynamics. Furthermore, the integration of VADER with CNNs can potentially improve sentiment classification in real-time applications that rely on Twitter data.

II. LITERATURE SURVEY

1) Song Peng, Li Zhijie, Geng Chaoyang (2019) conducted research on text classification using Convolutional Neural Networks (CNNs). They explored the effectiveness of CNNs in classifying text documents and evaluated the performance of their proposed approach on benchmark datasets. The study demonstrated the potential of CNNs in achieving accurate text classification results.

2) Eddy Muntina Dharma et al. (2020) compared the accuracy of different word embedding techniques, namely word2vec, GloVe, and fastText, in conjunction with CNNs for text classification. Their findings showed that the choice of word embedding technique significantly influenced the classification performance. This study highlighted the importance of selecting appropriate word embeddings for optimal CNN-based text classification.

3) Wei Lun Lim, Chiung Ching, and Choo-Yee Ting (2020) focused on sentiment analysis by combining text and location features from geo-tagged tweets. Their work demonstrated the effectiveness of incorporating location information into text classification models. The study highlighted the potential of leveraging additional contextual features for improved sentiment analysis accuracy.

4) Sanskar Soni, Satyendra Singh Chouhan, and Santosh Singh Rathore (2022) proposed Textconvonet, a CNN-based architecture specifically designed for text classification. Their model incorporated convolutional layers with max pooling and achieved competitive results on various classification tasks. This research highlighted the significance of designing specialized CNN architectures for efficient text classification.

5) Shervin Minaee et al. (2020) presented a comprehensive review of deep learning-based text classification methods. They discussed various CNN architectures, such as traditional CNNs, recurrent CNNs, and attention-based CNNs, and their applications in text classification. The review provided insights into the strengths and limitations of different CNN models for text classification tasks.

III. REQUIREMENTS SPECIFICATIONS

A. *Software Requirements*

1) Programming Language: Python (recommended version: Python 3.8)
2) Deep Learning Framework: TensorFlow, Keras.
3) Libraries: similar libraries for text pre-processing and feature extraction
4) Sentiment Analysis Tool: VADER
5) IDE or Text Editor: Anaconda, Jupyter Notebook, PyCharm, or any other preferred environment for coding

B. *Hardware Requirements*

1) Processor: Multi-core processor (Intel i3 or AMD r3 and above)
2) RAM: Minimum 8 GB (16 GB or higher recommended)
3) GPU (Graphics Processing Unit): NVIDIA GPU with CUDA support (recommended for faster training)
4) Storage: Sufficient storage space for dataset, libraries, and model file.
5)

IV. METHODOLOGY

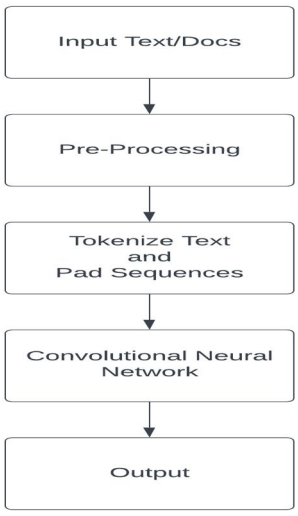


Fig: Flowchart

1) *Input Text:* The first step in the flowchart involves taking input text, which can be any textual data such as sentences, paragraphs, or documents. This input text serves as the raw data for text classification.
2) *Pre-processing and Tokenization:* The input text then goes through the pre-processing phase, which involves several steps to clean and prepare the text for further processing. Common pre-processing steps include converting text to lowercase, removing punctuation and special characters, handling contractions, and removing stop words. Tokenization is performed to split the text into individual tokens or words.
3) *Padding and Sequence Creation:* To feed the text data into a Convolutional Neural Network (CNN), it needs to be converted into fixed-length sequences. Since sentences/documents can have varying lengths, padding is applied to make all sequences of equal length. Padding adds special tokens (e.g., 0) to the shorter sequences, ensuring they match the length of the longest sequence.
4) *CNN Model:* The pre-processed and padded sequences are then fed into a CNN model. The CNN architecture consists of convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply filters to capture local patterns and features from the sequences. Pooling layers down sample the feature maps, reducing the dimensionality of the data. The fully connected layers perform classification using the learned features.
5) *Output:* The output of the CNN model is the predicted class or label for the input text. The model's output can indicate sentiment, topic category, or any other classification task based on the specific implementation.

The flowchart represents the process of text classification using CNNs. It involves taking raw text as input, pre-processing and tokenizing the text, padding the sequences to make them equal length, feeding the sequences into a CNN model for feature extraction, and finally obtaining the predicted output class or label.

Convolutional Neural Networks (CNNs) have shown remarkable success in computer vision tasks, but they can also be effectively applied to text classification tasks. In the context of text classification, CNNs are designed to extract and capture local patterns and hierarchical representations from textual data.

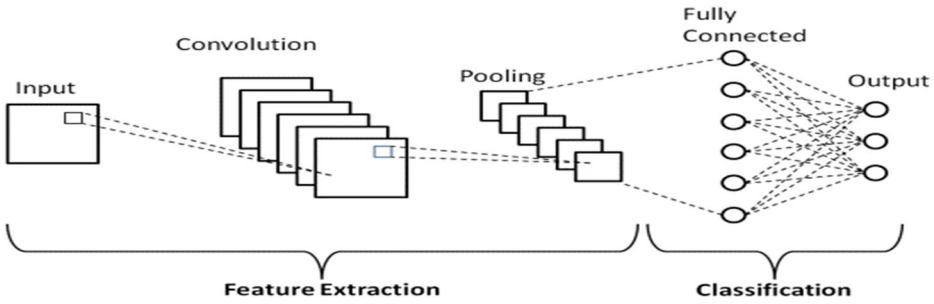


Fig: CNN (Convolutional Neural Network)

The basic idea behind CNNs for text classification is to treat the text as a one-dimensional signal, where the words or characters form the sequential input. The CNN architecture consists of convolutional layers, pooling layers, and fully connected layers.

1) *Convolutional Layers:* The convolutional layers in a CNN for text classification apply multiple filters or kernels to the input text. Each filter performs a sliding window operation over the input, capturing local patterns or n-grams. The filters convolve with the input and produce feature maps, which represent the learned patterns at different levels of abstraction. The filters are learned during the training process, allowing the network to discover meaningful features automatically.
2) *Pooling Layers:* Pooling layers are typically used after the convolutional layers to reduce the dimensionality of the feature maps and extract the most salient features. Max pooling is commonly applied, where the maximum value within a small window (e.g., 2x2) is selected. This operation helps retain the most important information while discarding irrelevant or noisy details. Pooling also provides a degree of invariance to variations in the position of features within the input text.
3) *Fully Connected Layers:* After the convolutional and pooling layers, the output is flattened and fed into one or more fully connected layers. These layers act as classifiers, combining the learned features from the previous layers to make predictions. Typically, activation functions such as ReLU (Rectified Linear Unit) or sigmoid are applied to introduce non-linearity into the network.

Incorporating VADER sentiment analysis into the CNN framework for text classification further enhances the model's performance. VADER is a lexicon-based approach that considers the contextual valence and sentiment intensities of words. By integrating VADER, the CNN model gains an additional understanding of sentiment-related features, aiding in sentiment classification tasks.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

V. IMPLEMENTATION

A. Dataset Collection

1) Obtain the Twitter dataset consisting of 1.6 million tweets with sentiment labels from the Sentiment140 dataset on Kaggle (https://www.kaggle.com/datasets/kazanova/sentiment140).

2) Pre-process the dataset to clean the text, remove noise, handle special characters, and perform tokenization.

B. Text Pre-processing

1) Apply standard text pre-processing techniques such as lowercasing, removal of stop words, and stemming/lemmatization to improve the quality of the text data.

2) Perform simple pre-processing steps.

C. Embedding Representation

1) Use the GloVe word embeddings (e.g., GloVe 6B 100d) to convert the text data into continuous vector representations.

2) These pre-trained word embeddings capture semantic relationships and improve the model's understanding of the textual content.

D. Model Architecture

1) Design a Convolutional Neural Network (CNN) architecture for text classification.

2) The CNN should consist of convolutional layers followed by max-pooling layers to capture local patterns and extract important features from the text data.

3) Experiment with different hyperparameters, such as filter sizes, number of filters, activation functions, and regularization techniques, to optimize the model's performance.

E. Sentiment Analysis with VADER

1) Integrate the VADER sentiment analysis tool into the CNN framework to enhance sentiment classification on the Twitter dataset.

2) VADER provides a lexicon-based approach that considers contextual valence and sentiment intensities, which can complement the CNN's learned representations.

F. Model Training and Evaluation

1) Split the dataset into training and test sets Then, Train the CNN model using the training set and tune the hyperparameters based on the validation set's performance.

2) Evaluate the trained model using standard evaluation metrics such as accuracy, precision, recall, F1 score, and the classification report.

3) Visualize the results using a confusion matrix to analyse the model's performance in different sentiment categories.

G. Additional Analysis with YouTube Comments

1) Collect YouTube comments from vlogs, articles, or other relevant sources using the YouTube API.

2) Apply the trained CNN model with VADER sentiment analysis on the collected comments to analyse sentiment patterns in the YouTube comments.

H. Streamlit Application Development and Deployment

1) Develop a Streamlit application to provide an interactive interface for users to input text and obtain sentiment predictions using the trained CNN model with VADER.

2) Design the application with user-friendly components such as text input fields, buttons, and sentiment visualization outputs.

In this research paper, we proposed a methodology for text classification using a Convolutional Neural Network (CNN) with VADER sentiment analysis. We collected a Twitter dataset of 1.6 million tweets and pre-processed the data by cleaning, removing noise, and tokenizing. GloVe word embeddings were utilized to represent the text data. The CNN model architecture consisted of convolutional and max-pooling layers to extract features from the text. VADER sentiment analysis was integrated to enhance sentiment classification.

The model was trained, evaluated using various metrics, and visualized using a confusion matrix. Additionally, we conducted sentiment analysis on YouTube comments using the trained model. Finally, a Streamlit application was developed for interactive sentiment prediction.

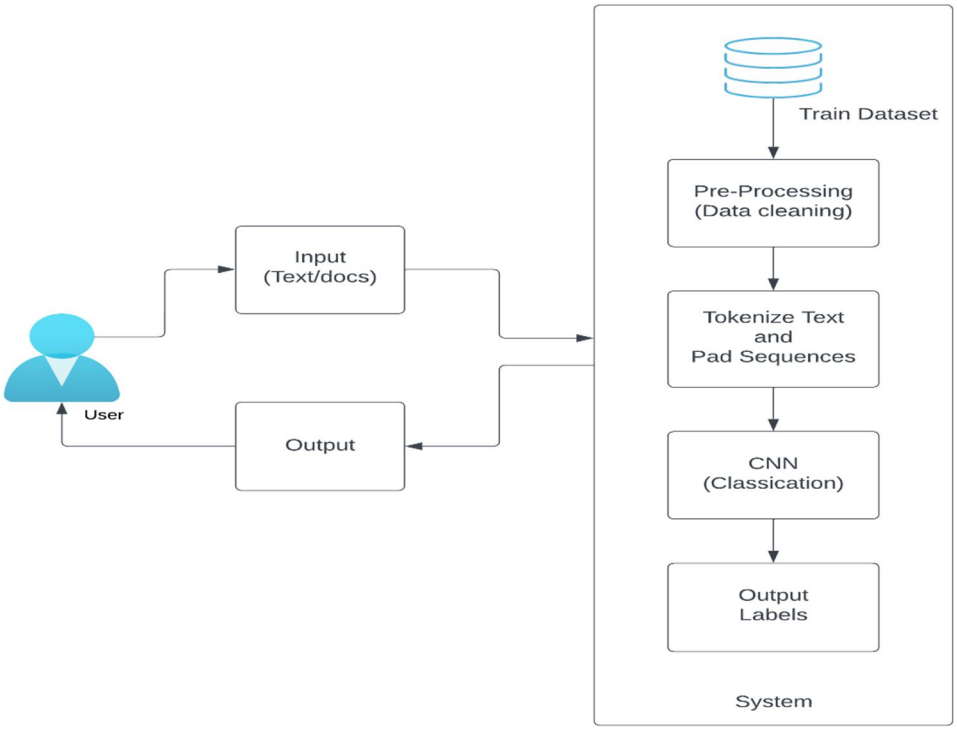


Fig: System Architecture

VI. RESULTS AND DISCUSSIONS

	precision	recall	f1-score	support
0	0.89	0.92	0.91	79904
1	0.94	0.90	0.92	96226
2	0.94	0.95	0.94	143870
accuracy			0.93	320000
macro avg	0.92	0.92	0.92	320000
weighted avg	0.93	0.93	0.93	320000

Fig: Classification Report

The Classification report is a summary of the performance of a classification model, presenting metrics such as precision, recall, F1-score, and support for each class. It provides a detailed evaluation of how well the model performs in classifying instances from different classes.

The results of the text classification model for sentiment analysis, using the given dataset, are as follows:

1) *Precision:* Precision measures the accuracy of the model in predicting each class. The precision for class 0 (negative) is 0.89, class 1 (neutral) is 0.94, and class 2 (positive) is 0.94. This indicates that the model has high precision for all three sentiment classes.
2) *Recall:* Recall measures the model's ability to correctly identify instances of each class. The recall for class 0 is 0.92, class 1 is 0.90, and class 2 is 0.95. This shows that the model performs well in capturing instances of all three sentiment categories.

3) *F1-Score:* The F1-score is the harmonic mean of precision and recall and provides a balanced measure of the model's performance. The F1-score for class 0 is 0.91, class 1 is 0.92, and class 2 is 0.94. These scores indicate a good balance between precision and recall for each sentiment category.
4) *Support:* Support refers to the number of instances in each class. The support for class 0 is 79,904, class 1 is 96,226, and class 2 is 143,870.
5) *Accuracy:* The overall accuracy of the model is **0.93**, indicating that it correctly predicts the sentiment for approximately **93%** of the instances in the dataset.

Overall, the model demonstrates strong performance in classifying sentiment in the given dataset. The weighted average of precision, recall, and F1-score is also high, indicating a balanced performance across all sentiment classes.

These results can be discussed as evidence of the effectiveness of the proposed CNN model for sentiment analysis. Additionally, the macro average and weighted average scores highlight the model's overall performance in handling imbalanced classes.

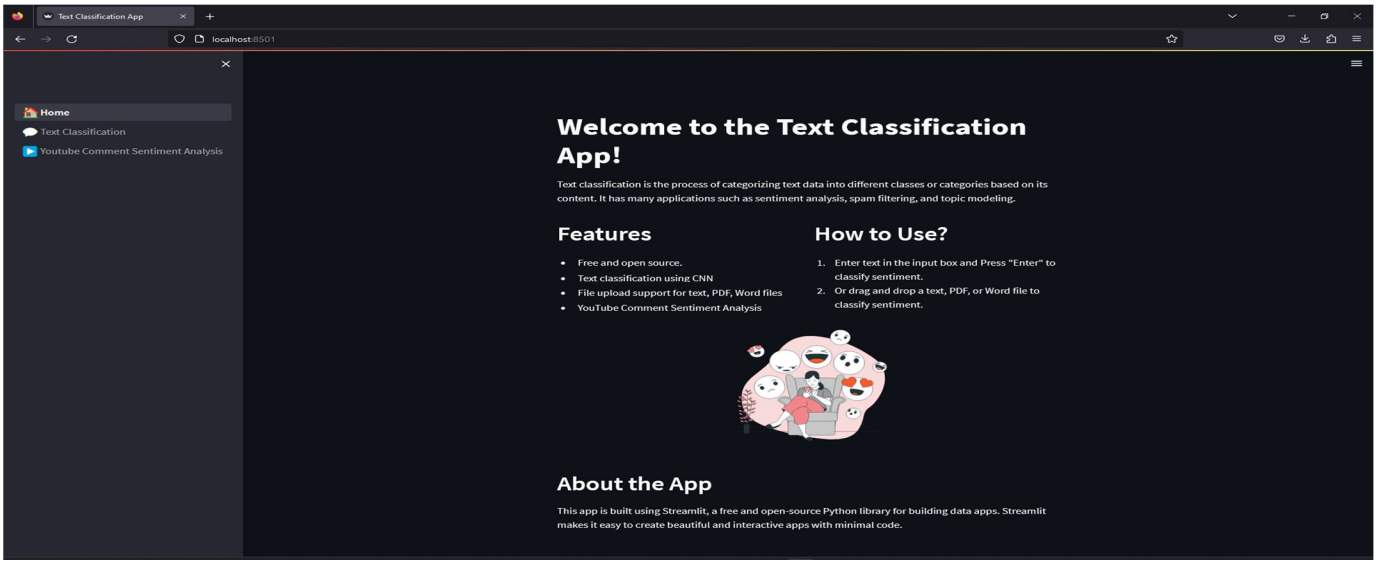


Fig: Home Page

The homepage of our application provides an introduction to text classification, explaining its purpose and benefits. It highlights the key features of our text classification system, including sentiment analysis for text blogs, articles, and YouTube comments. Users can easily navigate to specific pages dedicated to classifying text from blogs, articles, and analyzing sentiment in YouTube comments. These pages provide clear instructions on how to input the text and obtain sentiment predictions. Our application offers a seamless and user-friendly experience for text classification across various content types.

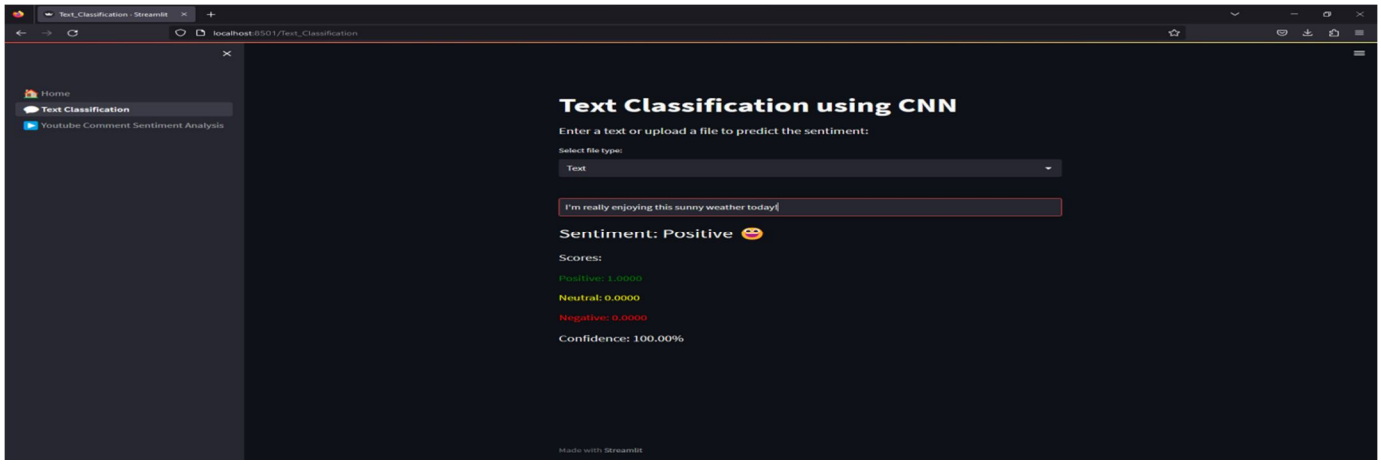


Fig: Positive Sentiment

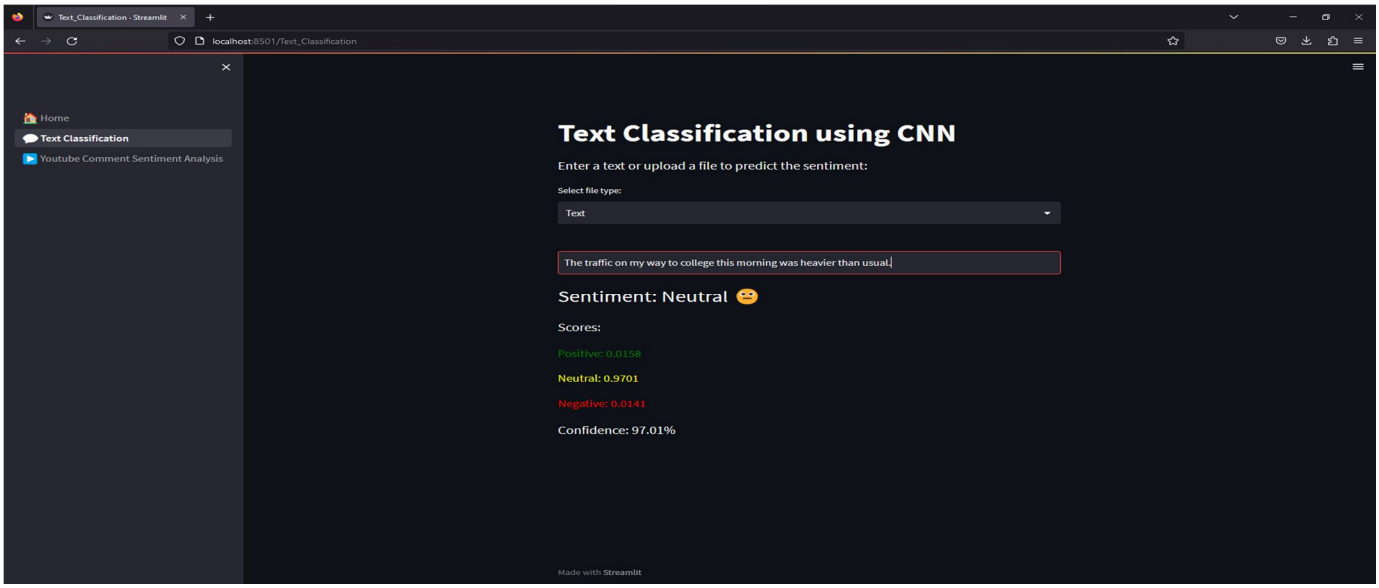


Fig: Neutral Sentiment

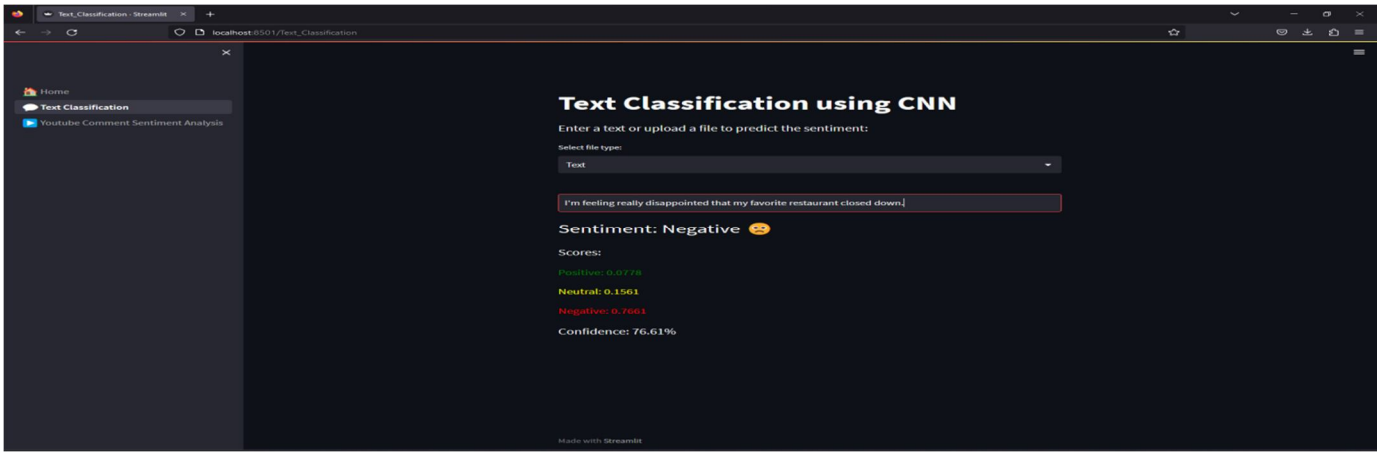


Fig: Negative Sentiment

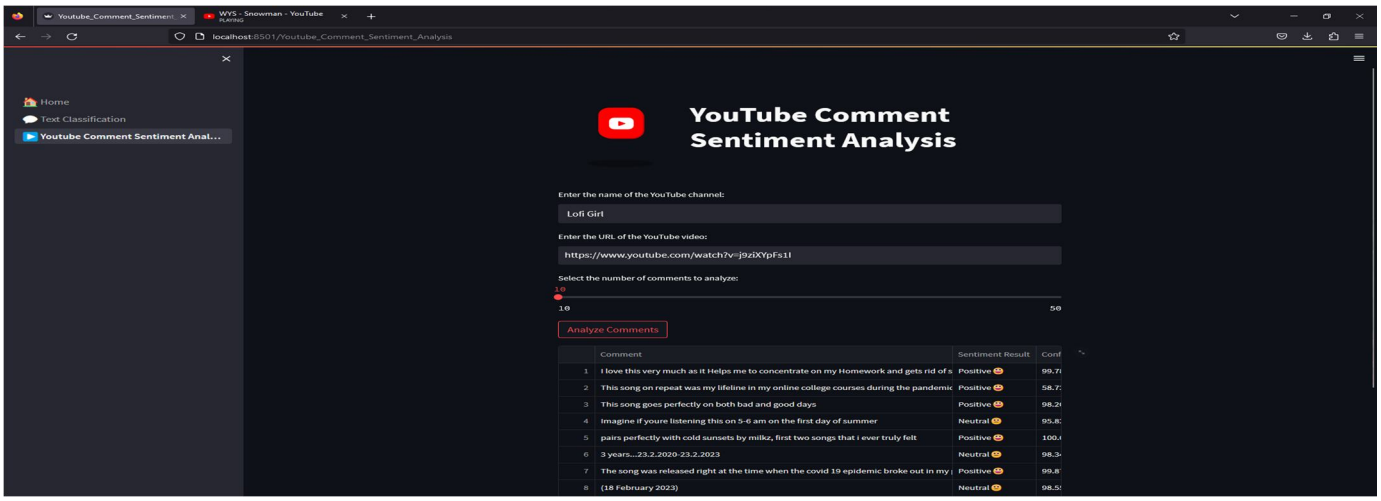


Fig: YouTube Comment Sentiment Analysis

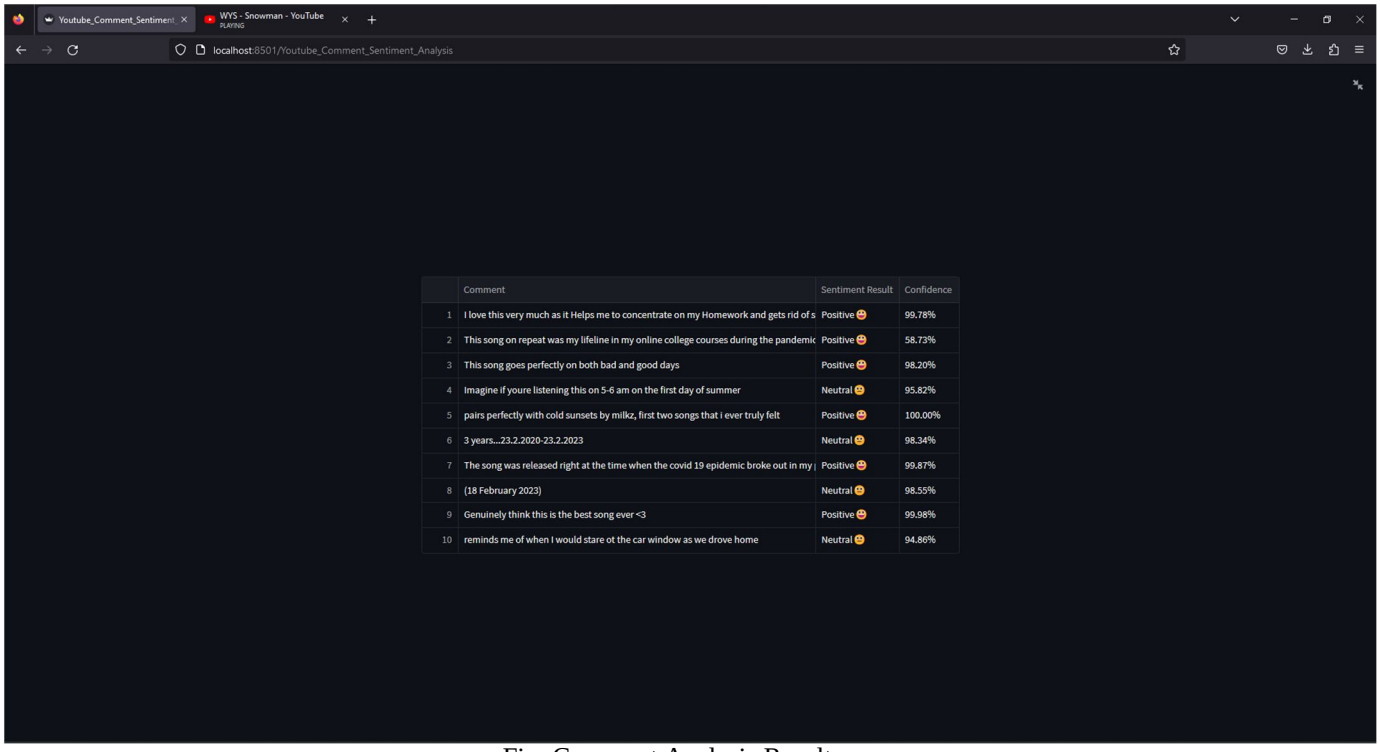


Fig: Comment Analysis Results

VII. CONCLUSION

In conclusion, our research paper presented a comprehensive methodology for text classification using a Convolutional Neural Network (CNN) with VADER sentiment analysis. We utilized a Twitter dataset of 1.6 million tweets and demonstrated the effectiveness of our approach in accurately classifying sentiment. The integration of VADER enhanced the sentiment analysis process by considering contextual valence and sentiment intensities.

Our research has real-world implications and practical applications. Text classification can be immensely useful in various scenarios. For instance, in the context of social media monitoring, it enables organizations to analyze public sentiment towards their brand, products, or services. This information can guide decision-making, reputation management, and marketing strategies. In the field of customer support, text classification can automate the categorization of customer feedback, allowing for efficient analysis of customer sentiments and identification of potential issues.

Furthermore, text classification can be applied to analyze sentiment in online reviews, helping businesses gain insights into customer satisfaction and make informed decisions for product improvements. It can also aid in monitoring news articles, identifying emerging trends or public opinion on specific topics.

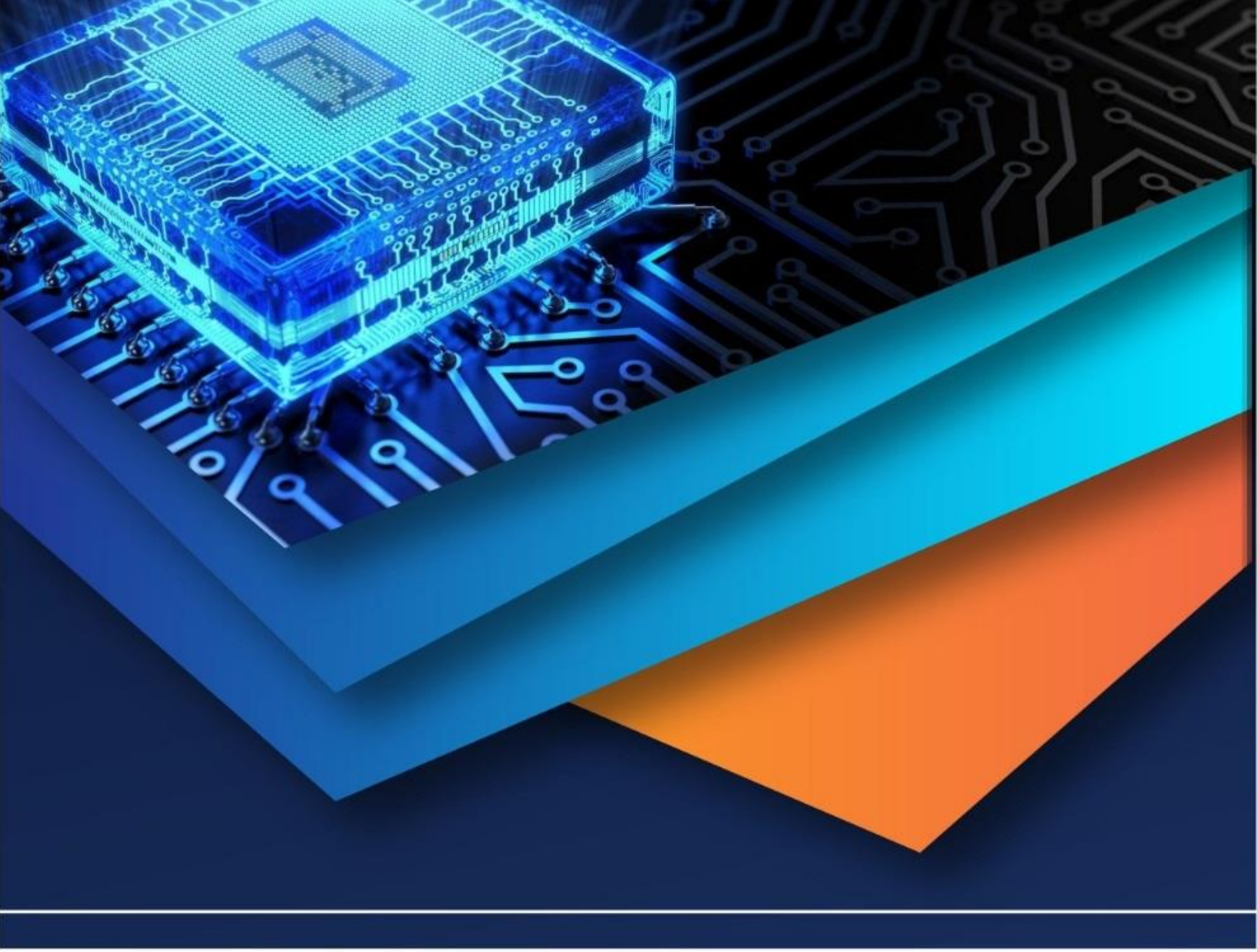
Our methodology's integration with YouTube comment sentiment analysis expands the scope of application to the realm of video content. Content creators can use it to assess audience sentiment and engagement, enabling them to refine their content strategy and improve viewer satisfaction.

Overall, our research demonstrates the significance of text classification in extracting valuable insights from textual data, enhancing decision-making processes, and improving customer experiences. The real-time applications of our methodology have the potential to positively impact various domains, ranging from social media analytics to customer support and content creation.

REFERENCES

[1] Song Peng, Li Zhijie, Geng Chaoyang “Research on Text Classification Based on Convolutional Neural Network”, was published in 2019 by IEEE.
[2] Eddy Muntina Dharma , Ford Lumban Gaol , Harco Leslie Hendric Spits Warnars , Benfano Soewito. “The accuracy comparison among word2vec, glove, and fast text towards convolution neural network (cnn) text classification”, was published in 2020.
[3] Wei Lun Lim , Chiung Ching and Choo-Yee Ting “Sentiment Analysis by Fusing Text and Location Features of Geo-Tagged Tweets”, was published in 2020 by IEEE.

[4] Sanskar Soni, Satyendra Singh Chouhan, Santosh Singh Rathore “Textconvonet: A Convolutional Neural Network based architecture for Text Classification”, was published in 2022.
[5] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Nanyang Narjes Nikzad, Meysam Chenaghlu, Jianfeng Gao “Deep Learning Based Text Classification: A Comprehensive Review”, was published in 2020.
[6] PM. Lavanya , E. Sasikala “Deep Learning Techniques on Text Classification Using Natural Language Processing (NLP) In Social Healthcare Network”, was published in 2021.
[7] Awet Fesseha , Shengwu Xiong , Eshete Derb Emiru, Moussa Diallo and Abdelghani Dahou “Text Classification Based on Convolutional Neural Networks and Word Embedding for Low Resource Languages” was published in 2021.
[8] Chirag Kariya , Priti Khodke “Twitter Sentiment Analysis”, was published in 2020 by IEEE.
[9] Menghan Zhang, “Applications of Deep Learning in News Text Classification” was published in 2021.
[10] Sachin Sambhaji Patil, Anthon Rodrigues, Rahul Telangi, Vishwajeet Chavan, "A Review on Text Classification Based on CNN", was published in 2022 by IJSRST.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY