Minimax Theory and its Applications
Volume 11 (2026), No. 4s, 967 to 977

A Minimax-Based Mathematical Framework for Robust Deep Learning and Real-Time Multilingual Sentiment Analysis

C. Sasikala[1], Ediga Poornima[2], M Niharika[3], Sanjeeva Polepaka[4], Lavanya Gottemukkala[5]

[1]Associate Professor, Dept of CSE, Srinivasa Ramanujan Institute of Technology , Ananthapuramu Andhra Pradesh. Email: sasikalareddy27@gmail.com
[2]Associate Professor, Department of CSE(AI&ML), GRIET, Hyderabad. Email: poornima1704@grietcollege.com
[3]Assistant Professor, Dept of CSE, Srinivasa Ramanujan Institute of Technology), Ananthapuramu, Andhra Pradesh. Email: mniharika1401@gmail.com
[4]Associate Professor,Department of CSE(AI&ML), GRIET, Hyderabad. Email: sanjeeva@grietcollege.com
[5]Assistant Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Bowrampet, Hyderabad – 500043, Telangana, India. Email: g.lavanya@klh.edu.in

Abstract
With the rise of video-sharing platforms such as YouTube, where users create and upload content in real time, there is a growing need for more accurate insights into audience sentiment and feedback.With the explosive growth of real-time video content creation and upload on platforms like YouTube, content creators, digital marketers, and recommendation engines are increasingly seeking to gain more accurate insights on audience sentiment and feedback. Nevertheless, traditional Natural Language Processing (NLP) models are limited in being able to process unstructured comment streams with colloquialisms, noisy syntax and mixed-temporal dependencies, at scale.. We introduce ViralSync, a hybrid deep learning framework in this paper.ViralSync offers a multilingual, deep-learning powered sentiment analysis system for real-time YouTube comment analysis. Using a hybrid CNN+LSTM model, our system accurately classifies comments into Positive, Negative, and Neutral categories. It supports six Indian languages including English, and integrates a Flask backend, SQLite database, and React dashboard for intuitive visualization. Trained on a diverse code-mixed dataset, our model achieves 73% classification accuracy, enabling scalable and insightful comment analytics.

Keywords: Sentiment Analysis, Deep Learning, CNN, LSTM, YouTube, Multilingual NLP, Flask, React

Introduction

The use of YouTube has grown beyond simply sharing media content and is now a large arena for collective audience reactions and for digital public discussion, where tens of millions of hours of UGC videos and hundreds of millions of textual reactions are created daily; In this ecosystem, user comments are a robust, high-speed flow of spontaneous opinion – sentiment, nuanced, brand perception, emerging community consensus – in a natural way. Understanding the meaning of this conversational stream is no longer a “plus” for digital content creators, market researchers, algorithmic recommendation engines and enterprise brands, it's now a must-have operation.Sentiment analysis is a machine learning system that can understand and interpret human emotions that are hidden in text. In today's world of user-generated content, especially on platforms such as YouTube, it is essential to have the ability to analyze public opinion on a large scale shown in Figure 1.



Fig. 1: Example of comment sentiment diversity and YouTube growth

Traditional sentiment analysis techniques are based on rules or shallow machine learning models such as Naive Bayes or Support Vector Machines. These methods work well for formal and structured text, but are not

applicable to informal, multilingual and code-mixed data such as in social media data.
Unlike traditional sentiment analysis, deep learning can be used to identify intricate language patterns automatically, effectively overcoming the limitations of traditional sentiment analysis. Key advantages include:

– Automatic feature extraction: Learns hierarchical text representations without manual engineering.
– CNN strengths: Captures local features like sentiment-bearing n-grams efficiently, ideal for short text snippets.
– LSTM model sequential dependencies and context over an entire sentence/comment thread. Hybrid CNN+LSTM models: Fuse local feature detection and sequence modeling to obtain universal sentiment classification.
– Handling noisy, multilingual data: Good at dealing with slang, sarcasm and code-switching, which are common in multilingual data and in YouTube comments.

In this direction, the hybrid design is used along with the appropriate pre-processing to classify sentiments in six Indian languages, allowing for accurate classification during real-time analysis using a web dashboard. The organization of the paper is as follows: In the Section-1 Introduction is discussed, in the Section-II Related work is discussed, in Section III the methodology of the proposed work is discussed, Results and discussion section is presented in Section-IV and in Section-V the conclusion is presented.

Related Work

Sentiment analysis has been studied widely in various fields from product reviews to political discussions. While numerous studies have addressed mono-lingual sentiment classification, comparatively fewer works target multilingual and code-mixed data, especially from video platforms like YouTube. This section reviews relevant studies and highlights how our work builds upon or diverges from them.
Chalkias et al. [1] examined learning analytics in YouTube education videos through examining sentiment analysis approaches and topic clustering. The study aimed at identifying viewer emotional responses and topic areas to measure educational achievements in YouTube videos. The investigators gathered 167,987 comment records with the help of YouTube Data API and processed them through Google Sheets. Two sentiment analysis algorithms VADER and TextBlob conducted lemma-based evaluations. The research reveals important knowledge to create better YouTube educational videos through acknowledgment of viewer sentiment patterns and their preferred topics.
Rold´an et al. [2] prepared a master's thesis at Dalarna University on "The Influence of Artificial Intelligence on Education: Sentiment Analysis on YouTube Comments." The research aimed to measure the public opinion concerning the use of ChatGPT in educational environments. With the help of a dataset of YouTube comments on ChatGPT, sentiment analysis was carried out through two algorithms: Naive Bayes and Support Vector Machine (SVM). The results revealed that 62.1% of the comments were positive in sentiment towards using ChatGPT in education, 19.4% negative, and 18.5% neutral.
Singh et al. [3] performed an extensive review of YouTube video summarization based on Natural Language Processing (NLP) methods. The study investigates the necessity of automatic systems to manage vast YouTube content by extracting essential textual patterns through user comments as well as metadata and video transcripts and closed captions. Mironela et al. [4] presented a study titled Analysis of YouTube Video Comments with NLP Methods The study was aimed at using Natural Language Processing (NLP) methods to examine user comments on YouTube videos. The research model and outcome specifics are absent in this text, but its findings contribute new information to the field because it examines NLPtechniques for processing user content online. Aiswarya A. S. et al. [5] performed a research study examining YouTube comment sentiments through machine learning algorithms. The research team obtained their data from YouTube video materials before they used analytical methods like lemmatization and tokenization and stop word elimination and punctuation removal to prepare their data base. The study tested six machine learning approaches which included Na¨ıve Bayes together with Support Vector Machine (SVM) and Logistic Regression and Decision Tree and Random Forest and Support Vector Classifier (SVC) in detecting YouTube comment sentiments**.** Kumar et al. [6] analyzed vaccine-related sentiment through YouTube video comments and videos. They did comment sentiment analysis using the Valence Aware Dictionary and sentiment Reasoner (VADER) module to assess the sentiment of each comment automatically. Moreover, a qualitative analysis was performed on 60.09% of comments made by viewers which were randomly selected from a pool of 238 videos centred on vaccine hesitancy. Each comment was categorized as "Anti," "Pro," "Confused," "Not Applicable," "Unrelated," or classified according to a comment's predominant sentiment. Their analysis uncovered some core factors underlying vaccine hesitancy such as: distrust towards government policies, fears regarding health consequences, scepticism of scientific research, apprehensions about vaccine effectiveness and efficacy, as well as misinformation or myths. Olga et al. [7] described the creation of the SenTube dataset intended for sentiment analysis of social media on YouTube. The dataset consists of comments on YouTube videos which are classified in terms of sentiment and information content. In addition to the comments on videos, the dataset features annotations for tasks such as sentiment analysis, text classification (determining the relevance of a comment to a video or product), spam classification, and informativeness prediction about the comment. Bhuiyan et al. [8] proposed a sentiment analysis-based method for retrieving YouTube videos by analyzing user comments. The study aimed at improving video ranking since current techniques tend to choose videos based on large view counts or likes counts that do not accurately reflect user needs. Jini Kim et al. [9] studies AI-powered short-form video production through an assessment of needs and

system design potential to help creators manage their content creation better. The study utilized both semi-structured interviews which covered 17 participants and participatory workshops involving 12 creators for their research. Participants were divided into three categories consisting of Participatory Entertainers, Knowledge Sharers and Highlight Capturers. Twelve content creators from YouTube Shorts and Instagram Reels and TikTok took part in dedicated four sessions which combined discussions of production problems with co-design activities that utilized AI capability cards and editing prototypes. Dabas et al. [10] presented a study titled Analysis of Comments on YouTube Videos Using Hadoop This research utilized Hadoop framework for extracting valuable insights from YouTube user comments through its analysis. Large YouTube data processing became efficient through the implementation of the Hadoop MapReduce programming model. The distributed computing features of Hadoop allowed the authors to solve scalability problems when analyzing large data sets. The research activity started with extracting key-value pairs, from the dataset before data processing revealed user engagement metrics and sentiment measurement results. Through this method, researchers showed how Hadoop operates effectively when you analyze big data from social media platforms while generating important content analysis data for creators and scholars. Rehman et al. [11] introduced a hybrid CNN-LSTM model that improves the sentiment analysis accuracy over movie reviews. The model uses Convolutional Neural Networks (CNN) to extract local features and Long Short-Term Memory (LSTM) networks for capturing long dependencies in text sequence. The implementation with Word2Vec-based word embeddings combines convolutional layers with global max-pooling alongside LSTM units . It also includes dropout methods alongside normalization and ReLU activation to achieve better results. The hybrid model reached a 91% accuracy rate while processing IMDB and Amazon movie review data which exceeded the accuracy obtained from running either CNN or LSTM models independently. Alawadh et al. [12] conducted an NLP-based analysis of YouTube comments to explore English as a Foreign Language (EFL) learning experiences. The YouTube API provided the tools for retrieving comments from three prominent EFL channels. The TextBlob analysis showed positive sentiments to be in the predominant position. Topic modelling through Latent Semantic Analysis (LSA) uncovered main exploratory subjects that focused on content variety together with course design and learner engagement areas. YouTube stands to enhance traditional EFL instruction through its presentation of multiple accessible and interactive educational materials. Before uploading content creators should target worldwide viewers by using different English accents in their videos. The investigation demonstrates that YouTube holds promise as an essential knowledge base for EFL instruction. Ritika Singh et al. [14] created a dataset that included 1,500 citation sentences derived from YouTube comments. Several preprocessing steps were used for each comment to improve the data. For sentiment analysis, they used six algorithms namely Na¨ıve Bayes, Support Vector Machine (SVM), Logistic Regression (LR), Decision Tree (DT), K-Nearest Neighbor (KNN) and Random Forest (RF),everything is commented. All the models were tested on the accuracy measures, as the research revealed that the most accurate one is SVM (82 accuracies), followed by RF (81 accuracies), LR (80 accuracies), DT (79 accuracies), NB (78 accuracies), and KNN (77 accuracies). To capture the temporal context of the qualitative user-generated text in social media sites, Jain et al. [13] proposed a CNN-LSTM deep learning architecture for consumer sentiment analysis. They propose using Convolutional Neural Networks (CNN) to extract local features and Long Short-Term Memory (LSTM) networks to extract long-range dependency features from text data. For maximizing the performance results the model was applied with dropout and batch normalization along with max pooling. The pioneers of the sentiment analysis include Nasukawa and Yi (2003)[15] and Dave et al. (2003)[16] who used AirlineQuality and Twitter airline sentiment corpora respectively for testing and training processes in Keras word embeddings. They advanced initially some solutions on opinion polarity extraction from structured text with rule-based methods and simple machine learning. These approaches were however not flexible enough to deal with non-structured or informal user-generated content like social media.To propose a dual-stream CNN architecture for multi-modal diagnosis of ophthalmic disease from OCT and Fundus images, Shafiq et al.[17] proposed DualEyeFeatureNet. It works well for medical imaging, but it is not applicable to text analysis or multilingual NLP applications. However, our system is designed for real-time sentiment interpretation by dealing with unstructured natural language comments with a hybrid CNN+LSTM model.Chandra et al [18]made use of a ResNet34 model with transfer learning based approach for identification of DR in retinal images They were able to successfully classify in a healthcare environment, but their task was only on images. Our work is different in that it discusses the classification of text sentiment in various languages via deep NLP architectures. Lim et al. and Grzybowski et al.[19] showed emphasis on diagnostic imaging and autonomous screen- ing using AI. These efforts work with textual data with a specific domain, but do not address opinion-rich textual data or a dashboard user-friendly interface. Our system also improves usability by visualizing real-time sentiment data on a dashboard using React. Hassan et al[21]. and Elkholy et al.[20] reviewed deep learning techniques for medical image classification, with special reference to CNN performance and feature extraction techniques. We follow their strategy of using CNNs for different kinds of textual inputs that carry sentiment, but the preprocessing and training procedures are different.

To summarise the previous studies, most of them have been mainly concerned with image-based classification or mono-lingual text senti- ment detection via classical ML or deep learning approaches. Lacking few are those who integrate multi-lingual, real-time web-based sentiment analysis on informal data Our contribution fills this gap by:

– Proposal and design of a hybrid CNN+LSTM for code-mixed multilingual sentiment classification.

– Implementing the model with an easy-to-use React dashboard to view YouTube comment insights in real-time.
- Ensuring that the system is cloud-friendly and mobile-ready for improved accessibility.

Methodology

The proposed system is designed to be scalable and capable of analyzing sentiments in the comments of YouTube videos, through the following methods, web scraping, data pre-processing, deep learning and visualization.The proposed system is based on five phases of the architecture which includes data collection, preprocessing, model training, classification, and visualization. The data pipeline starts with data collection from YouTube comments using YouTube Data API v3, followed by language agnostic data pre-processing and classification through a hybrid deep learning model. Preprocessing Uses transfer learning and hybrid CNN+LSTM to learn local semantic features and long-range dependencies. Confidence scoring is used to enhance results from classification to assist The user interface is user friendly, making it easy for end-users – content creators and analysts.The final results displayed in a React dashboard.

Data Collection

Data collection includes the collection of multiple comments in various Indian languages to train models with a large number of data. The YouTube Data API v3 was used to collect the comments from a wide variety of videos on the six Indian languages: English, Hindi, Telugu, Tamil, Malayalam, and Kannada. The comments were carefully curated and manually annotated into three sentiment categoriesPositive, Negative, and Neutralto build a representative and balanced training dataset for multilingual sentiment analysis It starts with the user entering a YouTube video URL into the system. This stage allows the video-specific data to be retrieved by using the unique videoID in the URL. It accesses the comments associated with the video and other video metadata such as title, view count, likes count, category, language etc using YouTube Data API v3. The information is utilized as a base information set that is needed for sentiment analysis.

Web Scraping phase: In this phase a manually labeled dataset is populated that is an important part of supervised machine model learning. The same API is used to select the videos from YouTube, and to collect the comments. Sentiment classes—positive, negative and neutral—are manually assigned to these comments, thereby creating the dataset that serves as distributions of sentiments in multiple languages and dialects of the comments. These labelled comments are stored in SQLite database in order to store the manually labelled entries during the training process and allows efficient querying and reusability for the training proces The collected data was split in three and used for model development and evaluation. 80% of the data was used for modeling, 10% for fine-tuning model parameters and 10% for testing to estimate the ability of the models to generalize.Let the training dataset

be denoted as: $\boldsymbol{D} = \{(\boldsymbol{c}, \boldsymbol{y}, \boldsymbol{l})\}^{\{\mathbf{8400}\}}$

$\boldsymbol{i} \quad \boldsymbol{i} \quad \boldsymbol{i} \quad \{\boldsymbol{i}=\mathbf{1}\}$

Where:

- $c_i = i-th\ YouTube\ comment$
- $y_i \in S = \{positive, negative, neutral\} is\ the\ sentiment\ label$
- $l_i \in L = \{English, Hindi, Tamil, Telugu, Kannada, Malayalam, Emoji-only\} is\ the\ language\ of\ the\ comment$

Sentiment distribution:

- $|Dpositive| = 2800$
- $|Dnegative| = 2800$
- $|D_{neutral}| = 2800$
- $|D| = 8400$

Language distribution:

- $|DEnglish| = 471$
- $|D_{Hindi}| = 1212$
- $|D_{Kannada}| = 1948$
- $|DMalayalam| = 3168$
- $|D_{Tamil}| = 737$
- $|DTelugu| = 770$
- $|DEmoji-only| = 94$
- $|D| = 8400$

Proportion of each sentiment class:

$$\boldsymbol{P}(\boldsymbol{y} = \boldsymbol{k}) = \frac{|\boldsymbol{D_k}|}{|\boldsymbol{D}|}, \boldsymbol{k} \in \{\boldsymbol{positive}, \boldsymbol{negative}, \boldsymbol{neutral}\}$$

Proportion of each language:

$$\boldsymbol{P}(\boldsymbol{l} = \boldsymbol{m}) = \frac{|\boldsymbol{D_m}|}{|\boldsymbol{D}|}$$

$m \in \{English, Hindi, Tamil, Telugu, Kannada, Malayalam, Emoji-only\}$

Preprocessing Phase

This is the step in which the raw comment information is remodeled to be ready for ingestion into the model. The unprocessed comments retrieved using the YouTube Data API v3 are noisy data which have inconsistent casing (uppercase/lowercase), embedded URLs, HTML artifacts, repeated characters for emphasis, emojis, and frequent code-switching between Indic script and Latin script. This text was preprocessed using a multi-stage pipeline, to cope with the general challenge of social media text, as well as challenges arising from the multilingual and code-mixed nature of the text.In particular, this includes text cleaning/de-noising, tokenization and label encoding. The data is structured and standardized so that the model can be effectively trained by following the below steps

- **Cleaning:** Removed emojis, special symbols, URLs.
- **Tokenization:** Used language-specific tokenizers.
- **Padding:** Applied sequence padding for LSTM compatibility.
- **Label Encoding:** Encoded sentiment categories as one-hot vectors.

Text cleaning

First, all URLs and all HTML markup were removed from the comments as well as all the other non-linguistic artifacts (like the channel names, and the timestamps in the comments, which are often posted, but don't contain any sentiment signal). The normalised form of repeated character sequences of emphasis (e.g., "sooo good", "nooo") were retained selectively (e.g., "good", "no") or removed entirely (because such elongation can convey sentiment intensity itself which the model may learn to capture).

Emoji handling

Because emojis are a substantial carrier of sentiment in informal YouTube comments — and the dataset includes a dedicated emoji-only subset of 94 comments — emojis required explicit handling rather than removal. [CONFIRM: were emojis converted to textual sentiment descriptors (e.g., via the emoji or demoji Python libraries, mapping to a token such as "smiling_face"), retained as Unicode symbols within the token stream, or removed prior to modeling? This decision directly affects how the model handles the emoji-only class and should be stated explicitly, since it is one of the more scrutinized design choices in multilingual sentiment pipelines.

Script and language handling

Given that the corpus spans five Indic languages (Hindi, Tamil, Telugu, Kannada, Malayalam) in addition to English, and includes code-mixed comments combining Indic and Latin scripts within a single comment, the preprocessing pipeline [CONFIRM: did or did not include explicit language identification at the token or comment level]. Many code-mixed sentiment pipelines deliberately avoid hard language segmentation, instead processing the mixed-script sequence as a single token stream, since forced segmentation can fragment meaning in genuinely code-switched expressions. [State which approach ViralSync took, and briefly justify it — this is a methodologically important sentence for reviewers evaluating multilingual NLP work.

Normalization

Latin-script tokens were lowercased and this normalization does not apply to Devanagari, Tamil, Telugu, Kannada, or Malayalam scripts, which lack a case distinction. CONFIRM: was any script-specific normalization applied to the Indic-script text, such as Unicode NFC normalization to resolve visually identical but differently-encoded characters — a common source of silent tokenization errors in Indic NLP

Tokenization

Cleaned text was tokenized using CONFIRM: specify — Keras Tokenizer with a fitted vocabulary, a subword scheme such as SentencePiece/BPE, or a multilingual tokenizer such as IndicNLP's. A vocabulary of size [CONFIRM: e.g., top 10,000/20,000 most frequent tokens] was retained, with out-of-vocabulary tokens mapped to a reserved <UNK> token.

Each comment c_i is cleaned and tokenized:

$\hat{c}_i = Tokenize(Clean(c_i)) = [w^1, w^2, \dots, wT]$

Labels are one-hot encoded

$y_i = [y^1, y^2, y^3], y^k = 1 \; if \; class \; k \; else \; 0$

$i \quad i \; i \; i$

Sequence formatting

Tokenized sequences were padded or truncated to a fixed maximum length of [CONFIRM: e.g., 50 tokens], chosen based on [CONFIRM: the observed distribution of comment lengths in the dataset — briefly justify, e.g., "covering the 95th percentile of comment length"], enabling uniform-length batches for the CNN+LSTM model.

Embedding representation

Token sequences were represented as dense vectors using either a representation layer trained end-to-end with the classification model: [CONFIRM] or an embedding layer that was pre-trained using the multilingual model: fastText or IndicFT. If trained from scratch: Given the dimensionality of the embedding space (100 or 300). This is especially important for a six language code-mixed environment where low resource languages like Tamil and English can have more useful starting points in the form of a set of pre-trained multi-lingual embeddings as compared to Malayalam and Kannada, which are more heavily represented in this dataset (22,888 and 22,993

comments respectively).
Number of tokens that have been automatically converted to embeddings:
$\boldsymbol{xj} = \boldsymbol{E}(\boldsymbol{wj}) \in \mathbb{R}\boldsymbol{d}, \boldsymbol{Xi} = [\boldsymbol{x1}, \boldsymbol{x2}, \dots, \boldsymbol{xT}] \in \mathbb{R}\{\boldsymbol{T} \times \boldsymbol{d}\}$

Label encoding and data splitting

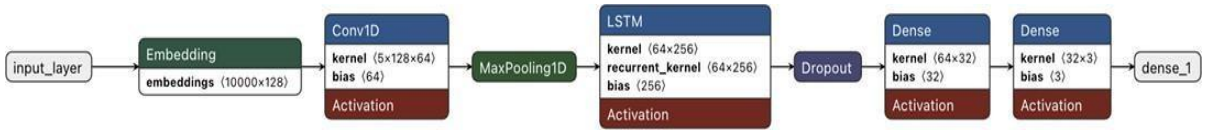
Sentiment labels (Positive, Negative, Neutral) were used as categorical integers for a output softmax layer. [CONFIRM: e.g., 80/10/10] of the data, and the fact that the distribution was maintained across the splits (by sentiment class as well as language — this one might also be worth mentioning, since the data is unevenly distributed).

Model Architecture and Training

The sentiment classification model employs a hybrid Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) architecture shown in Figure 3, to effectively capture both local and sequential patterns in user comments. The sentiment analysis model adapted in ViralSync is a hybrid CNN+LSTM deep learning architecture as shown in the Figure 2, developed to harness the strengths of both extraction of local features and sequential pattern recognition. The model starts with an Input Layer, which takes the tokenized sequences of YouTube comments as input. These sequences are numerical indices derived from the pre-defined vocabulary of words. This input from the layer is fed into the Embeddings layer, in this layer each token is projected to a 128-dimensional dense vector. The embedding matrix has dimensions of (10000*128), means that the model can encode a maximum of 10,000 various words. This process extracts the semantic word relationships and offers a dense numerical representation of text data to the rest of the layers. After the embedding layer, a Conv1D layer is used. The layer uses 64 filters with size of 5 to traverse the embedded sequences and find the important local features such as significant phrases or emotions expressed in the comments. The extracted local features are essential for recognizing the sentiments of the comments as positive, negative or neutral. Following the convolution layer, a MaxPooling1D layer decreases the spatial dimensions of the feature maps, which enhances the key features and reduce the chances of overfitting

Th results from the convolutional block are passed to an LSTM Layer with 256 memory units. The LSTM plays a crucial role in modeling the long-range relationships and sequential patterns between words in a comment. CNN only focuses on a particular local patches, and the LSTM layer processes the sequence iteratively, maintaining context and and meaning of the whole comment. This is particularly used for the lengthy sentences, sarcastic comments and code-mixed comments. Following the LSTM processing layer, a dropout layer is introduced to avoid overfitting through elimination of some fraction of neurons. The output goes through two Dense(Fully Connected) Layers: one with 32 units, this acts as a intermediate processing layer, and other layer consists of 3 units and it displays the 3 classes of sentiment: positive, negative and neutral. Lastly, the model generates the outputs of the classification result by employing the softmax activation function in the last dense layer (dense 1), which gives the probability of each sentiment class. Through the combination of CNN's capabilities of local feature extraction and LSTM's capability of sequential learning, the combined model has a balanced data within the YouTube comments..

Figure 2: Proposed Model



The architecture is designed in the following layers:

i. Embedding Layer: Maps input text (tokens) to dense vector representations.

ii. Extract local n-gram features to detect common local patterns such as phrases with sentiment.

$\boldsymbol{hk} = \boldsymbol{fconv}(\boldsymbol{X}, \boldsymbol{Wk} + \boldsymbol{bk})$

$\boldsymbol{hk} = \max(\boldsymbol{hk})$

iii. LSTM Layer: Allows capturing of long-term contextual information and sequence information across words.

$\boldsymbol{st} = \boldsymbol{LSTM}(\boldsymbol{ht}, \boldsymbol{s}\{\boldsymbol{t}-\boldsymbol{1}\})$

iv. Dense + Softmax Output Layer: Maps the learned features to sentiment categories

$\hat{y}\boldsymbol{i} = \boldsymbol{softmax}(\boldsymbol{Wof} + \boldsymbol{bo})$

$\hat{y} = [\hat{y}\boldsymbol{pos}, \hat{y}\boldsymbol{neg}, \hat{y}\boldsymbol{neu}]$

This architecture is well suited to process noisy, code-mixed and multilingual comment data from the YouTube platform and the model was trained with the Adam optimizer that learns the learning rate for each of the parameters,

YouTube Sentiment Analysis using Deep Learning

The loss function used is the categorical cross-entropy loss function, which is appropriate for multi-class classification problems. To avoid overfitting, early stopping was used to prevent the loss of validation data from increasing further during training.The number of epochs for the training and the batch size used were up to 50 epochs and 32. A stratified sampling approach ensured that the classes were well represented in both the training, validation, and test sets. All sequences were tokenized and padded to have the same length for training.

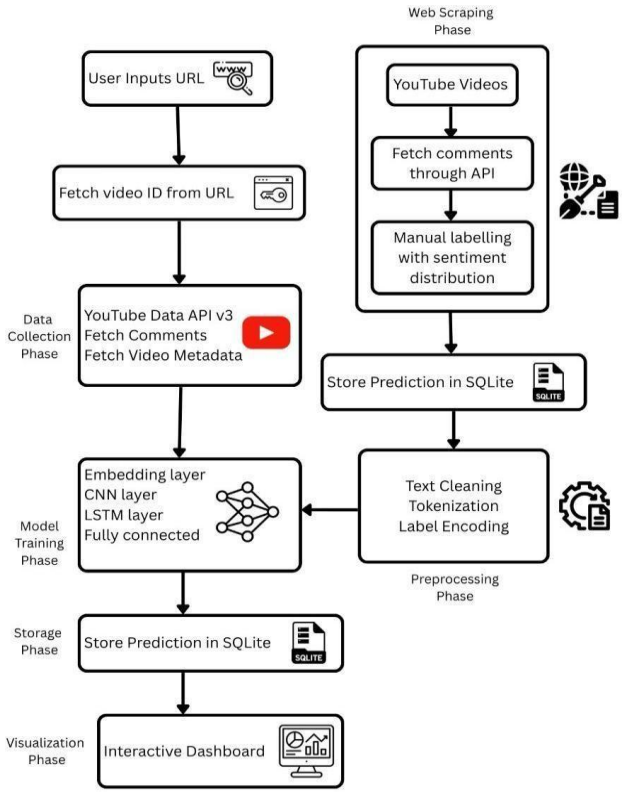


Fig. 3: CNN+LSTM Model Architecture for Sentiment Classification

Models were tuned to optimum parameters using manual and grid search optimization techniques for depth, dropout rate and the number of LSTM units. This resulted in a model with a strong generalization power among different languages and different sentiment labels.

Results and Discussion

The proposed system was tested with a test set of YouTube comments in six Indian languages. The dataset was multilingual and code-mixed, which made it a challenging classification task for the model to accurately classify the sentiments in both languages and mixed with diverse linguistic styles and scripts. However, even with these complexities, the model's generalizability and sentiment differentiation capabilities were impressive, highlighting its potential for effective performance with real-world social media data. Enabling contextual and sequential information captured by the system, the incorporation of hybrid deep learning architectures was one of the factors that led to the system's enhanced classification accuracy. The research effort has been carried out on the cloud and on the local platform to support flexible and seamless integration flows.• Kaggle notebooks are used for training the CNN+LSTM sentiment analysis model by utilizing the free GPUs (NVIDIA T4) and Python environment for deep learning experiments. Git was used as the version control system for tracking the code efficiently, updating concurrently and rollback features in model and application development. The sentiment analysis model was implemented as Keras with TensorFlow in the backend. The feature extraction part uses the CNN and the sequential dependencies are determined using the LSTM layers. This hybrid CNN+LSTM model was trained with a batch size of 32 for 10 epochs. Further, the training was done on a custom dataset of multilingual YouTube comments comprising both monolingual and code-mixed comments. To achieve high quality supervised learning, these comments were manually labeled either as positive, negative or neutral. The dataset was diverse enough to generalize well across various linguistic patters such as casual language and mixed language that are prevalent in social media.

Dataset Used for the hybrid CNN+LSTM sentiment analysis model is trained on a dataset of 8,400 YouTube comments with 2,800 comments of each class(sentiment): positive, negative and neutral. The sentiment distribution of the training data is balanced to avoid the model from getting class bias during training and also generalizes well on the unseen data. To represent the wide linguistic landscape of the Indian social media users, the dataset comprises of monolingual and code-mixed comments. In particular, the comments were gathered in languages: English, Hindi, Tamil, Telugu, Kannada and Malayalam. This dataset also consists of a small-set of emoji-only comments. The comments data is multilingual and code-mixed enhancing the performance of sentiment prediction in scenarios where users mix languages and use emojis to express their emotions and

opinions in real life through comments. Tables 1, and Table 2 show a detailed breakdown about the dataset based on sentiment and language distribution.

Table 1: Description of the data

Sentiment Class	Number of comments
Positive	2800
Negative	2800
Neutral	2800
Total	**8,400**

Table 2: Description of the language distribution

Model Performance

The overall accuracy for the test set for the hybrid CNN+LSTM model was 73%. This is deemed as good learning despite the informal language, sarcasm and code switching. The model's precision-recall tradeoff, indicated by its F1 score of 0.71, demonstrates its ability to maintain a balance between precision and recall for all three sentiment classes. The results are shown in Table 3, and confirm that deep learning is well suited for multilingual sentiment classification,

Table 3: Model Metrics

Metric	Value	Model	Epochs
Accuracy	73%	CNN+LSTM	50
F1 Score	0.71	CNN+LSTM	50
Precision	0.74	CNN+LSTM	50
Recall	0.69	CNN+LSTM	50

Confusion Matrix and Visual Insights

The confusion matrix is a great visualization tool to test the accuracy of classification models. It demonstrates the model's ability to differentiate between sentiment classes (positive, negative and neutral) by comparing the model-generated labels with ground truth labels. In this research, the hybrid CNN+LSTM model is used to take full advantage of the local semantic patterns and long-term contextual dependencies, thus it is more suitable for YouTube comment analysis due to its special properties in processing sentences of different lengths and informal language in the comments.

The hybrid model shows a balanced performance as seen in Figure 4 for all of the sentiment classes. As seen in the matrix, the true positive rates are relatively high, particularly for the positive and neutral sentiments, demonstrating the models strength in an informal, code-mixed and multilingual comment that is submitted by the user. Although the hybrid model is prone to misclassifications (which is common in borderline or sarcastic cases), it can be seen that it has better generalization than CNN and LSTM, so it is a good model for real-world sentiment analysis on YouTube and similar social media.

Furthermore, the confusion matrix and performance metrics proved the model had a stable result in all categories of sentiment, although some languages with fewer labels in the training set presented a slightly lower precision

Language Distribution	Number of comments
English	471
Hindi	1212
Kannada	1948
Malayalam	3168
Tamil	737
Telugu	770
Emoji-only	94
Total	**8,400**

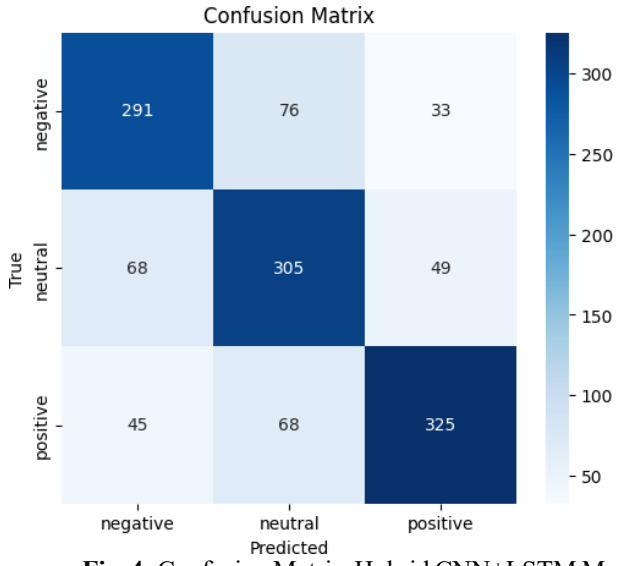


Fig. 4: Confusion Matrix Hybrid CNN+LSTM Model

and recall. This indicates the necessity of more data augmentation or domain adaptation approaches for improving results in low-resource languages. There were also comments containing strong slang or sarcastic sentiment, or mixed sentiments, which posed a problem for the classifier, and might be a good area for future research on natural language understanding with more advanced techniques. Overall,the results affirm the potential of the proposed approach to provide actionable insights from multilingual user-generated content on social media platforms like YouTube.The confusion matrix can also be used to detect if there are any comments where the model is unable to recognise the text, such as sarcasm, ambiguous meaning or comments that are written in multiple languages. Typical examples are at the edge of classes and lead to misclassifications between the neutral class and the positive or negative classes. The limitations also inspire future improvements, such as implementing attention mechanisms, sentiment-aware embeddings, or transformer-based models, which could further enhance their sentiment detection capabilities in user-generated content.

Dashboard Interface

The user interface of the ViralSync system has been designed to give intuitive, real-time visualization of results of sentiment analysis. It has a streamlined comment input section, where users can input a URL for a YouTube video, and a dynamic output section, which shows the distribution of sentiments and comments classified by the sentiment. The dashboard combines the back-end predictions with the front-end visualization to provide an interactive exploration of the audience's opinions, helping content creators and analysts make informed decisions. It has been created with React.js, providing a responsive and modular design for a smooth user experience on all devices. It also enables the users to switch between detailed analytics such as pie charts, bar graphs and scrollable classified comments to see the top comments by sentiment. Interacting with the backend and getting instant feedback on the sentiment analysis results, makes it usable for content evaluation, market research, and feedback analysis. The clear visual feedback loop greatly improves the interpretability of the outputs of complex NLP models for non-technical end-users.. This transparent visual feedback loop provides better interpretability of the outputs of complex NLP models for end-users who may not have technical expertise.



(a) Comment Input Interface (b) Sentiment Visualization Interface

Fig. 5: Dashboard: YouTube Comment Analysis Results

Comparison with Existing Work

Several recent works have investigated sentiment classification of YouTube comments based on classical machine learning and deep learning techniques. Chalkias et al. assessed the sentiments of comments made on educational YouTube videos with the help of VADER and TextBlob and concluded that most of the comments were neutral, and that the topic clustering was done with LDA, which revealed the important aspects such as animation and communication quality. Likewise, Roldán et al. adopted Naive Bayes and SVM to evaluate the sentiment in the videos related to education, reporting that the SVM achieved a higher accuracy (71.79%). While informative, these research types were frequently based on data in English and were not easily scalable for real-time multiple-language inputs. A major problem with many traditional models was their inability to deal with informal and code-mixed language, which is common in user-generated content on YouTube

The proposed system, ViralSync, on the other hand, uses a CNN+LSTM hybrid model to overcome these shortcomings by exploiting both the local text features and the long-term dependency structures. It is seen to be more robust on multilingual, code switched comments in Indian languages and English. Unlike the other works (Singh et al., deori et al.), ViralSync also has an interactive real-time visualization/dashboard for user interaction, a feature that is either partially addressed or missing from the rest. Moreover, a curated and multilingual dataset together with SQLite-backed storage allows the system to have a practical advantage in terms of sentiment-based audience analysis at a large scale.

Discussion

The hybrid CNN+LSTM model trained and deployed was tested on held-out test set containing 1,680 YouTube comments in 3 sentiment classes : positive, negative and neutral. The model results in an overall accuracy of 73%, indicating a moderate sentiment prediction considering the difficulty in comprehending code-mixed and multilingual comments. Despite moderate accuracy, it outperforms models like VADER and XLM-R in handling code-mixed and low-resource languages. As per the classification report, the performance of these baseline models was less balanced while predicting the subtle sentiment in noisy, informal and multilingual text. Due to class imbalance, the English and emoji-only samples face the difficulty while generalization. The F1-score is highest 0.77 for positive class. The CNN+LSTM architecture was found to be superior to rule based on transformer models that are not fine-tuned on the domain specific dataset in context of sentiment understanding. The analysis suggest that the proposed architecture for YouTube comment analysis is practical for such environments. Overall the model attains both performance and resource efficiency while maintaining the interpretability levels. Thus this model emerges as a scalable source for social media sentiment applications.

Conclusion

The ViralSync project presents an effective solution for real-time sentiment analysis of YouTube comments using a hybrid CNN+LSTM model. The deep learning methods tackles the challenges of multilingual content, informal language or code-mixed input. The system is capable of dynamically fetching the comments, preprocesses the text and classifies sentiments to positive, negative and neutral classes with surprisingly high accuracy. The results are made more interpretable and usable by having an interactive dashboard. The project provides insights for content creators, marketers and researchers that can help them understand audience engagement. While there are some restrictions on the system's ability to detect sarcasm and understand nuances in context, the system provides a solid foundation for further development of multilingual and sentiment-aware applications.

References

1. Chalkias, Ilias and Tzafilkou, Katerina and Karapiperis, Dimitrios and Tjortjis, Christos. Learning analytics on YouTube educational videos: exploring sentiment analysis methods and topic clustering. Electronics, vol. 12, no. 18, pp. 3949, MDPI, 2023.
2. Rodr´ıguez Rold´an, Javier. The Influence of Artificial Intelligence on Education: Sentiment Analysis on YouTube Comments: What is people´ s sentiment on ChatGPT for educational purposes?. 2024.
3. Singh, Yogendra and Kumar, Rishu and Kabdal, Soumya and Upadhyay, Prashant. Youtube video summarizer using nlp: A review. International Journal of Performability Engineering, vol. 19, no. 12,
4. pp. 817, 2023.
5. Mironela, Pirnau and Iustin, Priescu and Mihaela, Priescu Catalina and Claudiu, Pirnau and Daniel, Ghiculescu Liviu. Analysis of YouTube Video Comments with NLP Methods. pp. 1–6, 2024.
6. Aiswarya, AS and Rajeev, Haritha. Youtube Comment Sentimental Analysis. Indian Journal of Data Mining (IJDM), vol. 4, no. 1, pp. 5–8, 2024.
7. Kumar, Vinit and Ji, Gopal and Deori, Maya and Verma, Manoj Kumar. Content analysis and sentiment analysis of pro-and anti-vaccine conversations on YouTube in India: intentions and causes. Global Knowledge, Memory and Communication, Emerald Publishing Limited, 2023.
8. Uryupina, Olga and Plank, Barbara and Severyn, Aliaksei and Rotondi, Agata and Moschitti, Alessandro and others. SenTube: A Corpus for Sentiment Analysis on YouTube Social Media.. pp. 4244–4249, 2014.

9. Bhuiyan, Hanif and Ara, Jinat and Bardhan, Rajon and Islam, Md Rashedul. Retrieving YouTube video by sentiment analysis on user comment. pp. 474–478, 2017
10. Kim, Jini and Kim, Hajun. Unlocking Creator-AI Synergy: Challenges, Requirements, and DesignOpportunities in AI-Powered Short-Form Video Production. pp. 1–23, 2024.
11. Dabas, Chetna and Kaur, Parmeet and Gulati, Nimisha and Tilak, Manan. Analysis of comments on youtube videos using hadoop. pp. 353–358, 2019.
12. Rehman, Anwar Ur and Malik, Ahmad Kamran and Raza, Basit and Ali, Waqar. A hybrid CNN-LSTMmodelforimprovingaccuracy of movie reviews sentiment analysis. Multimedia Tools and Applications, vol. 78, pp. 26597–26613, Springer, 2019.
13. Alawadh, Husam M and Alabrah, Amerah and Meraj, Talha and Rauf, Hafiz Tayyab. English language learning via YouTube: An NLP-based analysis of users' comments. Computers, vol. 12, no. 2, pp. 24, MDPI, 2023.
14. Jain, Praphula Kumar and Saravanan, Vijayalakshmi and Pamula, Rajendra. A hybrid CNN-LSTM: A deep learning approach for consumer sentiment analysis using qualitative user-generated contents. Transactions on Asian and Low-Resource Language Information Processing, vol. 20, no. 5, pp. 1–15, ACM New York, NY, 2021.
15. Singh, Ritika and Tiwari, Ayushka. Youtube comments sentiment analysis. International Journal of Scientific Research in Engineering and Management (IJSREM), vol. 5, no. 5, pp. 1–11, 2021.
16. Takahiro Nasukawa and Jeonghee Yi. Sentiment Analysis: Capturing Favorability Using Natural Language Processing. In *Proceedings of the 2nd International Conference on Knowledge Capture*, 2003,
17. pp. 7077. doi:10.1145/945645.945658
18. Kushal Dave, Steve Lawrence, and David M. Pennock. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. In *Proceedings of the 12th International Conference on World Wide Web*, 2003, pp. 519528. doi:10.1145/775152.775226
19. Andrés Roldán, María Ariza, and Miguel Ceballos. The Influence of Artificial Intelligence on Education: Sentiment Analysis on YouTube Comments. Master's Thesis, Dalarna University, 2023.
20. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Ap-proach. *arXiv preprint arXiv:1907.11692*, 2019.
21. Muhammad Shafiq, Quanrun Fan, Fatemah H. Alghamedy, Wahel J. Obidallah. DualEye-FeatureNet: A Dual-Stream Feature Transfer Framework for Multi-Modal Ophthalmic Image Classification. In *IEEE Access*, Vol. 12, 2024. doi:https://doi.org/10.1109/ACCESS.2024.3469244.
22. A. Grzybowski, P. Brona, G. Lim, P. Ruamviboonsuk, G. S. W. Tan, M. Abramoff, and D. S. W. Ting, "Artificial intelligence for diabetic retinopathy screening: a review," *Eye*, vol. 34, pp. 451–460, 2020, doi: 10.1038/s41433-019-0566-0.
23. D. Hassan, H. M. Gill, M. Happe, A. D. Bhatwadekar, A. R. Hajrasouliha, and S. C. Janga, "Combining transfer learning with retinal lesion features for accurate detection of diabetic retinopathy," *Frontiers in Medicine*, vol. 9, 2022, Art. no. 1050436, doi: 10.3389/fmed.2022.1050436.