https://ejournal.svgacademy.org/index.php/ijiee/

#### **OPEN ACCESS**



Vishal Kumar Jaiswal<sup>1</sup>, Chrisoline Sarah J<sup>2</sup>, T. Harikala<sup>3</sup>, K. Reddy Madhavi<sup>4</sup>, M. Sudhakara<sup>5</sup>

**Original Article** 

<sup>1</sup>Sr. Manager Software Engineering, OPTUM, Ashburn, Virginia, 20148.

<sup>2</sup>Assistant Professor, Department of Mathematics, New Horizon Institute of Technology and Management, Thane West, Maharashtra. <sup>3</sup>Assistant Professor, Dept of CSE, Annamacharya University, Rajampeta, India.

<sup>4</sup>Professor, Department of AI&ML, School of Computing, Mohan Babu University, Tirupati, A.P., India.

<sup>5</sup>Senior Computer Vision Scientist, ChiStats Labs Private Limited, Pune.

\*Corresponding Author's Email: vishal.jaiswal@optum.com, chrisolinesarah@nhitm.ac.in, hariekalla@gmail.com, kreddymadhavi@gmail.com, mallasudhakar.cse@gmail.com

# Abstract

Emotion identification from textual data is crucial for understanding human emotions in applications such as user behaviour analysis, targeted content delivery, and mental health monitoring. Even though English has advanced significantly, emotion detection in Hindi, a widely spoken language lacking enough resources, is still tricky because of data, grammatical complexity, and NLP tools. This study presents a deep neural network (DNN)-based framework for emotion classification in Hindi sentences. Our approach includes Hindi-specific preprocessing using the iNLTK library, comparative evaluation of multiple encoding techniques (Bag of Words, TF-IDF, Word2Vec), and training a robust DNN model to classify text into five emotion classes: joy, sadness, anger, suspense, and neutral. Experimental results on the BHAV dataset demonstrate that our DNN model achieves a balanced accuracy of 94.91%, outperforming traditional classifiers such as Naive Bayes, SVC, Logistic Regression, Decision Trees, and Boosted Trees. The confusion matrix and training-validation curves confirm the model's generalization capabilities and minimal overfitting. Our outcomes underscore the significance of deep learning in low-resource language settings and set the groundwork for future improvements in multimodal emotion detection, code-mixed data handling, and deployment in real-time Hindi NLP applications.

Keywords – Emotion Detection, Hindi Text, Deep Neural Network (DNN), Natural Language Processing (NLP), Text Classification, Low-Resource Language, SMOTE, Machine Learning.

# Introduction

Due to its numerous uses in fields including sentiment analysis, social media analytics, human-computer interaction, mental health monitoring, and targeted content delivery, emotion detection from textual data has accelerated significantly in recent years [1]. With increasing user engagement on digital platforms, people often express their feelings, intentions, and sentiments through informal texts, often in their native languages [2]. Among these, Hindi—spoken by over 600 million people—is one of the most widely used languages on the internet [3]. Despite this, emotion detection in Hindi textual data remains a relatively underexplored area.

Emotion detection is a subtask of affective computing that aims to classify text into predefined emotional categories such as anger, joy, sadness, fear, or disgust [4]. While substantial progress has been made in English and other resourcerich languages, models built for Hindi face several challenges [5]. The limited availability of high-quality annotated emotion datasets in Hindi restricts model training and evaluation [6]. Hindi exhibits rich morphology, free word order,



and complex syntax, making it harder for standard models to capture context. The increasing use of Hinglish (Hindi written in Latin script mixed with English) further complicates the task. The lack of professionally trained language models and NLP tools tailored to Hindi reduces the effectiveness of traditional NLP pipelines [8].

Traditional machine learning models, including naive bayes, Support Vector Machines (SVM), Logistic Regression, and Decision Trees, rely heavily on handcrafted features and bag-of-words representations, which fail to capture the deep semantic and contextual cues embedded in human emotions. Additionally, these models do not scale well when confronted with noisy, unstructured, or informal data prevalent in social media.

To bridge this gap, we present a Deep Neural Network (DNN)-based framework for accurate and scalable emotion detection in Hindi textual data. Our architecture leverages the representational power of deep learning to learn discriminative features from complex linguistic patterns automatically. The system follows a systematic and language-aware approach. We utilize the Indic NLP Library (iNLTK) to handle Hindi-specific preprocessing tasks, including tokenization, stopword removal, and text normalization. We evaluate multiple text vectorization strategies—ranging from classic Bag-of-Words (BoW) and TF-IDF to dense embeddings like Word2Vec and contextual embeddings like BERT—ensuring comprehensive semantic and syntactic information coverage. Our DNN architecture includes multiple fully connected layers with dropout regularization and ReLU activation, culminating in a softmax layer for multi-class emotion classification.

Our motivation stems from the following key observations and challenges: The growing presence of Hindi users on digital platforms requires language-specific emotion recognition tools. In existing models, cultural and emotional subtleties are frequently lost due to translation or disregard for non-English material. Although deep learning models have outperformed conventional techniques in comparable natural language processing tasks, Hindi emotion detection has not yet been sufficiently investigated.

The contribution of our work is as follows:

- 1. We design a preprocessing pipeline for Hindi language processing using iNLTK, addressing tokenization, stopword handling, and noise removal in native text.
- 2. We comprehensively compare text encoding techniques BoW, TF-IDF, and Word2Vec, to determine the optimal input representation for Hindi emotion classification.
- 3. We introduce a deep neural network that outperforms conventional machine learning models' accuracy and generalization on Hindi emotion datasets.
- 4. We validate our model using real-world Hindi datasets and benchmark it against traditional classifiers such as Naive Bayes, SVC, Logistic Regression, Decision Trees, and Boosted Trees.
- 5. We provide detailed insights into model behavior, including its robustness and overfitting tendencies, through confusion matrix analysis and training-validation performance visualization.

# Literature Survey

Cross-domain sentiment analysis has garnered significant attention, particularly in transfer learning and attention-based architectures. Manshu et al. [9] introduced the Hierarchical Attention Network with Prior Knowledge (HANP), which leverages emotion dictionary matches to detect crucial semantic pivots and non-pivots. In parallel, Huang et al. [10] expanded the capabilities of transformer-based models by developing FriendsBERT and ChatBERT—domain-specific BERT variants pre-trained on conversational datasets like EmotionLines [11], which includes dialogues from the *Friends* TV series and Facebook Messenger. Their approach achieved micro F1-scores of 81.5% and 88.5% on the respective subsets, indicating that pre-training on contextually relevant dialogue significantly improves emotion recognition in multi-turn conversations.

Polignano et al. [12] tested GoogleEmb, GloVeEmb, and FastTextEmb on datasets including ISEAR, SemEval 2018 Task 1, and SemEval 2019 Task 3 to investigate conventional word embeddings for emotion detection. Their approach



significantly outperformed the traditional SVM and Random Forest models, achieving an F1-score of up to 0.84 on SemEval 2018. This investigation validated the efficacy of optimized embeddings for emotion recognition in general textual settings.

Al-Azani and El-Alfy [13] used a multimodal technique for Arabic video sentiment analysis, combining textual, audio, and visual information in a hybrid framework. Using their own SADAM dataset, which was constructed from YouTube videos, the proposed fusion mechanism achieved 95.08% accuracy, demonstrating the efficacy of merging many modalities for deeper emotional comprehension. Compared to previous models, this represents an increase of nearly 6%.

In affective computing, Wang et al. [14] introduced a tree-structured CNN-LSTM model to predict Valence-Arousal (VA) scores using regional feature extraction. Evaluated on four datasets, including SST, EmoBank, and CVAT, the model attained a Pearson correlation of 0.809 on SST and varying but solid performance on other corpora, confirming its utility in continuous emotion estimation tasks.

Zhang et al. [15] have also examined knowledge-guided mechanisms, proposing the Knowledge-Guided Capsule Attention Network (KGCapsAN). This architecture integrates Bi-LSTM and attention mechanisms for sentiment analysis and was validated across six datasets. It achieved accuracies above 74% on Twitter, Lap14, and Rest datasets from SemEval, indicating the advantage of incorporating prior knowledge structures.

Lu et al. [16] tackled audio sentiment classification by coupling RNN-based classifiers with self-attention mechanisms. Using the SWBD dataset containing 140 hours of speech, they improved the state-of-the-art accuracy on IEMOCAP from 66.6% to 71.7%, demonstrating the benefit of ASR-based pretraining for audio-based sentiment understanding.

Lexicon-based techniques continue to provide interpretable sentiment analysis. Yin et al. [17] proposed the FCP-Lex method for constructing sentiment dictionaries using part-of-speech-based CP chunks. Applied to the LMRD and MRD datasets, this lexicon-based classifier achieved 82.10% and 71.34% accuracy, respectively, and outperformed standard lexicons in most benchmarks.

To address low-resource languages, Ahmad et al. [18] introduced a cross-lingual transfer model using English embeddings to detect emotions in Hindi news articles. The Emo-Dis-HI dataset created from disaster-related news was used alongside EmoSemEval-EN. Their model achieved an F1-score of 0.863 in English data but dropped to 0.53 in Hindi, reflecting challenges in domain transfer and linguistic divergence.

Twitter remains a popular platform for sentiment research. Matla and Badugu [19] analyzed tweets using Russell's Circumplex model and applied Naïve Bayes and k-NN classifiers. Using the Sentiment140 dataset, they obtained 72.6% accuracy with Naïve Bayes, suggesting that rule-based preprocessing coupled with eager learners can yield reasonable performance.

Rule-based approaches are also featured in the work of Seal et al. [20], who built a keyword-based classifier using the ISEAR dataset and achieved a 66.18% F1 score. The approach is appropriate for lightweight, explainable applications because of its simplicity and clarity despite its poor accuracy.

Seo et al. [21] introduced Heterogeneous Modality Transfer Learning (HMTL) as a final solution to the unimodal sentiment transfer gap—their model leverages adversarial learning to transfer knowledge from textual to audio-visual domains using CMU-MOSI and IEMOCAP datasets. The auditory and visual models achieved F1 scores of over 60%, demonstrating the feasibility of cross-modal transfer learning even in the absence of direct textual inputs.

# **Methods and Materials**

### A. Dataset Description

In this study, we employ the BHAAV dataset, India's first and most comprehensive Hindi-language dataset dedicated to emotion recognition in literary narratives, consisting of 20,304 sentences from 230 short stories spanning 18 thematic categories, including genres such as Inspirational and Mystery [22]. Each sentence is annotated with one of five



emotional labels—joy, suspense, anger, sadness, or neutral—based on how a typical reader might perceive the emotions conveyed by the characters.

Annotation was carried out manually by a group of native Hindi-speaking volunteers. The annotators' at least 10 years of formal Hindi education and evident love of literature ensured accuracy in language and interpretation.



Fig. 1: Distribution of the sentence in class

According to an initial examination of the dataset, the neutral group makes up about 60% of all utterances, which shows a notable class imbalance in Figure 1. This skewed distribution challenges most machine learning models, which assume a roughly equal representation across classes to optimize accuracy and generalization. When datasets are imbalanced, models tend to overfit to the majority class, resulting in high overall accuracy but poor performance on the minority categories—ironically undermining the primary objective of emotion classification.

To mitigate this issue, the Synthetic Minority Over-sampling Technique (SMOTE) was initially applied. As suggested by previous research [23], this method usually consists of two steps: first, utilizing random undersampling to reduce the size of the majority class and then using SMOTE to create synthetic samples for the minority classes to produce a balanced distribution. Figure 2 depicts the impact of this approach. Figure 2 (a) displays the class distribution before applying SMOTE, whereas Figure 2 (b) shows the results following SMOTE. However, as seen from the ongoing imbalance, the method did not produce satisfying outcomes in our instance. To remedy this, we manually oversampled the minority classes to get a more balanced dataset. This paper reports every subsequent experimental analysis based on the dataset's final version, shown in Figure 4.



Fig. 2: Dataset balancing using SMOTE





Fig. 3: Dataset balancing using manual class weight

### **B.** Input File Handling

The input file contains unprocessed Hindi sentences that must be classified into specific emotion categories. A custom Python script reads these sentences individually and stores them in a list. Each sentence is taken from the list and goes through a multi-step text preprocessing pipeline. This pipeline is designed to clean and prepare the data to ensure it is suitable for a classification model.

### C. Data Preprocessing

Preprocessing is a vital phase in the emotion classification pipeline, as raw text data often contains extraneous elements that can obscure meaningful patterns. We employed the iNLTK (Indic NLP Toolkit) library, which is optimized for Hindi language processing. The following steps were performed:

- Tokenization: Segmenting sentences into individual words or tokens.
- Punctuation Removal: Eliminating all punctuation symbols to reduce noise.
- Stop Word and Pronoun Removal: Filtering out common stop words and personal pronouns with limited semantic weight in emotion classification.

This cleaning process ensures that the resulting text is syntactically minimal but semantically rich, which improves the quality of features extracted in subsequent steps. This step is crucial when using frequency-based vectorization techniques like TF-IDF, where irrelevant tokens can bias feature distributions.

### D. Text Encoding

To make the textual data interpretable by machine learning models, it is essential to convert it into numerical format a process known as **text encoding**. We employed a diverse set of encoding strategies to ensure comprehensive feature representation:

• **Bag of Words (BoW)**: Represents each document by the frequency of words it contains. Formally:

BoW 
$$(w_i, d_i)f_{i,i}$$

Where,  $f_{i,i}$  defines the frequency of a word  $w_i$  in document  $d_i$ .

• Term Frequency-Inverse Document Frequency (TF-IDF): Enhances BoW by downscaling the weights of frequent terms across the corpus:

**TF-IDF** 
$$(w_i, d_j) = TF(w_i, d_j) \times \log(\frac{N}{DF(w_i)})$$

where, **TF-IDF**  $(w_i, d_j)$  is the term frequency, N is the total number of documents, and DF  $(w_j)$  is the number of documents containing  $w_i$ .



• Word2Vec Embedding: Learns dense vector representations of words in a semantic space, preserving contextual relationships.

Each of these encoding techniques generated a feature matrix that served as input to various machine learning classifiers. The multi-embedding strategy ensures robustness across emotion categories, particularly in a linguistically rich and morphologically complex language like Hindi.

#### E. Model Development

• Classification Framework Using Deep Neural Networks: We utilize a deep neural network (DNN) architecture optimized for multi-class sentiment classification to classify emotional expressions embedded in Hindi textual data. Deep learning models, intense neural networks, have shown superior performance over traditional machine learning techniques when extracting complex semantic features from natural language inputs. However, training deep models involves challenges such as vanishing gradients and overfitting due to increased model complexity. To address these, we integrate Batch Normalization, Rectified Linear Unit (ReLU) activation, and zero-padding within a hierarchical architecture. Figure 4 depicts the architecture of the DNN.



Fig. 4: Proposed deep neural network

• **Deep Neural Network Architecture:** Our DNN consists of dense layers interspersed with normalization and non-linear activation functions. The objective is to extract higher-order abstract representations from input Hindi text, which has been preprocessed using standard NLP techniques.

Let, the input Hindi text be represented as a fixed-length vector  $x \in \mathbb{R}^n$  generated from an embedding layer. The forward propagation for each dense layer l is defined as:

$$z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)}$$
  
$$a^{(l)} = \phi (BatchNorm (z^{(l)}))$$

where,  $W^{(l)}$  and  $b^{(l)}$  are the learnable weights and biases of layer l,  $\phi$  is the activation function,  $a^{(l-1)}$  is the output from the previous layer,

BatchNorm is the batch normalization operation defined as:

BatchNorm 
$$(z_i) = \gamma \left( \frac{z_i - \mu}{\sqrt{\sigma^2} + \epsilon} \right) + \beta$$

here,  $\mu$  and  $\sigma^2$  are the batch mean and variance respectively, and  $\gamma$ ,  $\beta$  are trainable scale and shift parameters.



The activation function  $\phi$  is chosen as Leaky ReLU, defined by:

$$\phi(\mathbf{x}) = \begin{cases} x & if \ x > 0 \\ ax & otherwise \end{cases}$$

where  $\alpha \in (0,1)$  is a small constant (typically 0.01) used to mitigate the **dying ReLU** problem.

• Output Layer and Emotion Classification: The final dense layer projects the last hidden activation  $a^{(L-1)}$  to the number of emotional classes C, followed by a softmax function to compute the class probabilities:

o = 
$$W^{(L)}a^{(L-1)} + b^{(L)}$$
  
 $\hat{y}_i = \frac{exp(o_i)}{\sum_{i=1}^{C} exp(o_i)}$  for i = 1,2,.....C

here,  $\hat{y}_i$  is the predicted probability of class i, and the model predicts the class with the highest probability.

• Loss Function and Optimization: We utilize the categorical cross-entropy as the loss function for multi-class classification:

$$\mathscr{L} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

where,  $y_i \in \{0,1\}$  is the ground-truth label for class i, and  $\hat{y}_i$  is the predicted probability.

The model is optimized using the **Adam optimizer**, which adapts the learning rate for each parameter using estimates of the first and second moments of the gradients:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \acute{\mathbf{m}}_t$$

where,  $\dot{m}_t$  and  $\dot{\upsilon}_t$  are bias-corrected estimates of the mean and uncentered variance of the gradients,  $\eta$  is the learning rate, and  $\theta$  are the model parameters.

• **Model Regularization and Dropout:** Dropout layers are added between dense layers to reduce overfitting and improve generalization. Each neuron is retained with a probability p, effectively making the network robust to noisy inputs:

$$a_{drop}^{(l)} = a^{(l)} \odot r, \tilde{r_i}$$
 Bernoulli (p)

Where,  $\bigcirc$  element-wise multiplication and r is a binary mask.

# **Results and Discussions**

#### A. Experimental Setup

The experiments were carried out using a balanced subset of the BHAAVis Hindi emotion classification dataset, comprising 20,304 sentences. The five emotion categories of joy, sadness, anger, suspense, and neutrality were equally represented in these phrases. An 80:20 split was used to separate the dataset into training and testing sets to ensure accurate analysis. The models were implemented using Python 3.9 and evaluated with Scikit-learn, TensorFlow 2.x, and the Keras API. Sensibly, to ensure practical calculation and reduced training periods, all training and evaluation operations were conducted on a computer system that included an Intel Core i7 CPU, 32 GB of RAM, 512 SSD and an NVIDIA RTX 3080 GPU.

TF-IDF vectors were utilized to construct input characteristics for conventional machine learning models. Deep learning models, on the other hand, used Word2Vec representations in conjunction with BERT embeddings to capture more detailed semantic information.



#### **B.** Evaluation Metrics

Various assessment metrics were used to evaluate the models' performance, including Accuracy, Precision, Recall, and F1-score. True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) are the terms utilized to describe the categorization results.

Accuracy: Accuracy measures the percentage of all properly predicted cases, which quantifies the model's total accuracy, which is computed as:

Accuracy: 
$$\frac{Tp+TN}{Tp+TN+FP+FN}$$

**Precision:** The precision of a prediction is the ratio of accurately predicted positive observations to all expected positive observations. In situations when the cost of false positives is considerable, it is especially pertinent.

Precision: 
$$\frac{TP}{TP+FP}$$

**Recall:** Recall estimates the model's ability to identify all relevant instances in the dataset correctly:

Recall: 
$$\frac{TP}{TP+FN}$$

**F1-Score:** When there is an unequal distribution of classes, this balanced metric, which is the harmonic mean of precision and recall, is used. When the expense of false positives and false negatives is about equal, it is advantageous:

F1-score: 2 
$$\frac{Precision.Recall}{Precision + Recall}$$

#### **B.** Performance of the Models

The performance of models and the proposed deep neural network (DNN) was evaluated on the balanced BHAAVis Hindi emotion classification dataset. Table 1 illustrates the evaluation outcomes based on **accuracy, precision, recall,** and **F1-score,** presenting a detailed comparison of model effectiveness.

Among the baseline models, Boosted Trees (XGBoost) achieved the highest performance, with an accuracy of 83.21% and an F1-score of 82.04%. This is attributable to its ensemble learning nature, which incorporates considerable weak learners to create a strong classifier. It consistently outperformed other traditional models by more effectively capturing complex patterns in the text data. The second-place Support Vector Classifier (SVC) accuracy was 81.03%. Its capacity to identify the best decision boundaries in high-dimensional feature spaces—particularly when employing BERT embeddings—probably contributed to its strong overall performance. Meanwhile, Logistic Regression achieved 79.68% accuracy, showing that even simple linear models can perform competitively when combined with strong text representations like TF-IDF and BERT.

Model	Accuracy	Precision	Recall	F1-score
Naive Bayes	74.32%	72.15%	73.08%	72.61%
Logistic Regression	79.68%	78.42%	77.91%	78.16%
Support Vector Classifier	81.03%	80.12%	79.67%	79.89%
Decision Tree Classifier	77.55%	75.98%	76.51%	76.24%
Boosted Trees (XGBoost)	83.21%	82.33%	81.76%	82.04%
Proposed DNN	95.48%	94.87%	95.31%	95.09%

Table 1: Performance of the models

Less appropriately, the DT Classifier and Naive Bayes models achieved accuracies of 77.55% and 74.32%, respectively. Although quick and easy to understand, these models have trouble capturing subtle language patterns, particularly in context-dependent, emotionally charged texts like Hindi phrases. A constraint in natural language problems where word dependencies are essential is the assumption of feature independence made by Naive Bayes. All baseline models were significantly outperformed by the suggested Deep Neural Network (DNN), which obtained an F1 score of 95.09% and an exceptional accuracy of 95.48%. This performance breakthrough is due to the model's ability to capture syntactic and semantic components utilizing Word2Vec and BERT-based embeddings, which offer rich, contextual text



representations. Furthermore, because of its layered design, the DNN can efficiently represent intricate hierarchical relationships in language that are difficult for conventional models to comprehend.

#### C. Confusion Matrix Analysis

A confusion matrix, illustrated in Figure 5, was developed to evaluate the proposed DNN performance for emotion classification further.



*Figure 5: Confusion matrix of the DNN model* 

- High Accuracy in Joy and Neutral: Since Joy and Neutral classes are more prevalent in the dataset, it is not surprising that the model does well in identifying them. The matrix's thick matching diagonal cells suggest little confusion with other classes.
- Moderate Confusion Between Sadness and Anger: Anger and sadness are mistakenly categorized in a notably high proportion of cases. This can explain the contextual resemblance and modest semantic overlap between these two emotional states in narrative texts.
- Suspense Classification Challenges: The *Suspense* class exhibits some misclassification, particularly with the *Neutral* and *Joy* classes. This suggests that the linguistic markers for suspense in Hindi texts may be less distinct or that suspense is often co-expressed with emotionally neutral or anticipatory language.
- Balanced Detection Across Categories: Despite the initial class imbalance in the raw dataset, the final balanced dataset and custom oversampling techniques ensured that the DNN could learn meaningful patterns across all emotion categories. The confusion matrix confirms that the model completely neglected the class.

Overall, the confusion matrix substantiates the robustness of the proposed DNN model in classifying complex emotional expressions in Hindi texts. While minor overlaps exist between semantically adjacent classes, the model maintains high predictive fidelity, especially in distinguishing between clearly defined emotional categories. Future work could further involve integrating context-aware attention mechanisms or transformer-based encoders to improve differentiation in closely related emotion classes.

#### **D.** Comparative Summary

Table 3 presents the accuracy, balanced accuracy, and F1-score of various classification models. The suggested Deep Neural Network (DNN) continuously outperformed the other models in every assessed measure. In particular, it handled the dataset with precision and dependability, achieving an accuracy of 95.48%, a balanced accuracy of 94.91%, and an F1-score of 95.09%. Despite being the most competitive conventional model, XGBoost still trailed the DNN by a



considerable margin, achieving an accuracy of 83.21% and an F1-score of 82.04%. On every performance metric, the DNN outperformed other models, such as LR and SVC, by about 13–16% despite their moderate efficacy.

Model	Accuracy	Balanced Accuracy	F1-score
Naive Bayes	74.32%	73.65%	72.61%
Logistic Regression	79.68%	78.94%	78.16%
Support Vector Classifier	81.03%	80.47%	79.89%
Decision Tree Classifier	77.55%	76.88%	76.24%
Boosted Trees (XGBoost)	83.21%	82.74%	82.04%
Proposed DNN	95.48%	94.91%	95.09%

Table 2: Accuracy, balanced accuracy and f-score of the models

The suggested Deep Neural Network (DNN) continuously outperformed the other models in every assessed measure. In particular, it handled the dataset with precision and dependability, achieving an accuracy of 95.48%, a balanced accuracy of 94.91%, and an F1-score of 95.09%. Despite being the most competitive conventional model, XGBoost still trailed the DNN by a considerable margin, achieving an accuracy of 83.21% and an F1-score of 82.04%. On every performance metric, the DNN outperformed other models, such as LR and SVC, by about 13–16% despite their moderate efficacy.



Figure 6: The accuracy, balanced accuracy, and F1-score of various classification models

The competing architectures RetinaNet, Faster R-CNN, and Cascade R-CNN also demonstrated competent performance, with mAP scores between 0.84 and 0.86. These results suggest that traditional region-based detectors perform reliably under a joint-class scenario, mainly due to their refined region proposal mechanisms and strong feature hierarchies. RetinaNet's focal loss function contributed to balanced learning despite class variations, enabling it to achieve slightly higher accuracy among the three.

However, EfficientDet's more advanced compound scaling and lightweight backbone allowed it to learn more generalized representations across all species, which is particularly useful when real-time or edge-computing requirements constrain model deployment. The significant boost in its mAP under a unified label also implies that EfficientDet can be an effective end-to-end detection tool in practical road safety systems where quick, reliable identification of large animals, regardless of species, is essential.

**D.** Real-Time Detection Efficiency and Computational Throughput: The efficiency of object detection models in realtime scenarios is as critical as their accuracy, particularly in automated driving systems where delays can lead to hazardous consequences. Table 5 compares the detection speeds of several state-of-the-art architectures. The fastest



model was RetinaNet R-50-FPN, which produced 50.6 FPS for one-class detection and 45.2 FPS for identifying all ten animal classes. Because of its competitive precision and speedy performance, it is a good choice for systems that require real-time responsiveness.

# **Conclusion and Future Work**

Identifying emotions in text has become essential to natural language processing, particularly for applications that need psychological assessment, sentiment analysis, and human-computer interaction. Even while high-resource languages like English have seen tremendous advancements, Hindi, a linguistically rich but under-resourced language, nevertheless has a difficult task when it comes to emotion identification. Hindi narratives have morphological variation, syntactic variability, and cultural complexity that necessitate a specific and contextually aware approach.

In this study, we suggested a deep neural network architecture created especially for Hindi text data emotion identification. Our method included a proprietary neural architecture, hybrid encoding approaches, and rigorous text preparation to capture emotional semantics across five emotion categories. Future work could explore transformerbased architectures like BERT or IndicBERT fine-tuned on Hindi corpora, integrate multimodal inputs such as audio and visual cues for richer emotion understanding, and expand the scope to include code-mixed and colloquial Hindi used in social media. Moreover, real-time deployment in educational and mental health applications presents an exciting avenue for practical impact.

### References

- Seshakagari, Haranadha Reddy Busireddy, et al. "Dynamic financial sentiment analysis and market forecasting through large language models." International Journal of Human Computations & Intelligence 4.1 (2025): 397-410.
- [2] M. Zhou, J. Ju, W. Yuan, L. Liu, and Y. Feng, "Exploring the roles of informational and emotional language in online government interactions to promote citizens' continuous participation," Public Performance & Management Review, vol. 48, no. 2, pp. 468–496, 2025.
- [3] Reddy, Busireddy Seshakagari Haranadha, R. Venkatramana, and L. Jayasree. "Enhancing apple fruit quality detection with augmented YOLOv3 deep learning algorithm." International Journal of Human Computations & Intelligence 4.1 (2025): 386-396.
- [4] Z. Liu, L. Qian, Q. Xie, J. Huang, K. Yang, and S. Ananiadou, "Mmaffben: A multilingual and multimodal affective analysis benchmark for evaluating llms and vlms," arXiv preprint arXiv:2505.24423, 2025.
- [5] P. Pakray, A. Gelbukh, and S. Bandyopadhyay, "Natural language processing applications for low-resource languages," Natural Language Processing, vol. 31, no. 2, pp. 183–197, 2025.
- [6] R. Mahajan, A. S. More, and U. Shah, "Navigating emotion in code-mixed languages: Performance of ml and dl models on hindi-english text," Procedia Computer Science, vol. 258, pp. 4029–4037, 2025.
- [7] M. Balipa, K. Anwaya, M. Murugappan et al., "A rule-basedmachine translation framework for low-resource language pairs," in 2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL). IEEE, 2025, pp. 969–974.
- [8] R. Raja and A. Vats, "Parallel corpora for machine translation in low-resource indic languages: A comprehensive review," arXiv preprint arXiv:2503.04797, 2025.
- [9] T. Manshu and W. Bing, "Adding prior knowledge in hierarchical attention neural network for cross domain sentiment classification," IEEE Access, vol. 7, pp. 32 578–32 588, 2019.



- [10]Y.-H. Huang, S.-R. Lee, M.-Y. Ma, Y.-H. Chen, Y.-W. Yu, and Y.-S. Chen, "Emotionx-idea: Emotion bert-an affectional model for conversation," arXiv preprint arXiv:1908.06264, 2019.
- [11]S.-Y. Chen, C.-C. Hsu, C.-C. Kuo, L.-W. Ku et al., "Emotionlines: An emotion corpus of multi-party conversations," arXiv preprint arXiv:1802.08379, 2018.
- [12]M. Polignano, P. Basile, M. de Gemmis, and G. Semeraro, "A comparison of word-embeddings in emotion detection from text using bilstm, cnn and self-attention," in Adjunct publication of the 27th conference on user modeling, adaptation and personalization, 2019, pp. 63–68.
- [13]S. Al-Azani and E.-S. M. El-Alfy, "Enhanced video analytics for sentiment analysis based on fusing textual, auditory and visual information," IEEE Access, vol. 8, pp. 136 843–136 857, 2020.
- [14]J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Tree-structured regional cnn-lstm model for dimensional sentiment analysis," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 581–591, 2019.
- [15]B. Zhang, X. Li, X. Xu, K.-C. Leung, Z. Chen, and Y. Ye, "Knowledge guided capsule attention network for aspect-based sentiment analysis," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 2538–2551, 2020.
- [16]Z. Lu, L. Cao, Y. Zhang, C.-C. Chiu, and J. Fan, "Speech sentiment analysis via pre-trained features from endto-end asr models," in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 7149–7153.
- [17]F. Yin, Y. Wang, J. Liu, and L. Lin, "The construction of sentiment lexicon based on context-dependent part-of-speech chunks for semantic disambiguation," IEEE Access, vol. 8, pp. 63 359–63 367, 2020.
- [18]Z. Ahmad, R. Jindal, A. Ekbal, and P. Bhattachharyya, "Borrow from rich cousin: transfer learning for emotion detection using cross lingual embedding," Expert Systems with Applications, vol. 139, p. 112851, 2020.
- [19]M. Suhasini and B. Srinivasu, "Emotion detection framework for twitter data using supervised classifiers," in Data Engineering and Communication Technology: Proceedings of 3rd ICDECT-2K19. Springer, 2020, pp. 565–576.
- [20]D. Seal, U. K. Roy, and R. Basak, "Sentence-level emotion detection from text based on semantic rules," in Information and Communication Technology for Sustainable Development: Proceedings of ICT4SD 2018. Springer, 2020, pp. 423–430.
- [21]S. Seo, S. Na, and J. Kim, "Hmtl: Heterogeneous modality transfer learning for audio-visual sentiment analysis," IEEE Access, vol. 8, pp. 140 426–140 437, 2020.
- [22]T. Kumar, M. Mahrishi, and G. Sharma, "Emotion recognition in hindi text using multilingual bert transformer," Multimedia Tools and Applications, vol. 82, no. 27, pp. 42 373–42 394, 2023.
- [23]N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," Journal of artificial intelligence research, vol. 16, pp. 321–357, 2002.

