Multi-modal Meme Troll and Domain classification

1st Aaryan Nijhawan
Artificial Intelligence, NITK
aaryannijhawan.211ai002@nitk.edu.in

2nd Ashitosh Phadatare
Artificial Intelligence, NITK
ashitoshphadatare.211ai007@nitk.edu.in

3rd Prathipati Jayanth
Artificial Intelligence, NITK
jayanthprathipati2003@gmail.com

***Abstract*—This project presents a holistic approach to meme trolling detection and domain classification, focusing on Telugu and Kannada languages. Leveraging a spectrum of methodologies ranging from basic machine learning models such as Support Vector Machines (SVM), Random Forest, Naive Bayes, to image-based models like Convolutional Neural Networks (CNN), ResNet-50, and state-of-the-art models such as CLIP, multilingual BERT, XLM-BERT, and Vision Transformers, we explore diverse modalities including image classification, extracted text classification, and combined text-caption classification. Our system integrates multiple models to achieve two primary goals: accurately detecting trolling behavior and classifying memes into thematic domains like politics, movies, sports.. By training on multilingual data and considering linguistic diversity, our approach ensures robust performance across different linguistic contexts, providing valuable insights into meme culture and trolling behavior in Telugu and Kannada-speaking communities.**

***Index Terms*—Multimodal memes, Caption Generation, Text Extraction, Image classification, Text Classification, m-BERT, XLM-BERT, CLIP.**

I. Introduction

Memes are a common form of communication in the modern digital age, appearing on social media sites and in online groups all over the world. Memes have a big impact on online conversation and reflect society trends because of their unique blend of comedy, satire, and cultural allusions. Nevertheless, despite their enormous appeal, memes are also used as platforms for the dissemination of false information, the support of extreme viewpoints, and online harassment. The vast amount of meme content that is produced and distributed on a daily basis makes content analysis and moderation extremely difficult. In response, our initiative seeks to better comprehend and tackle the complexity of meme culture by developing an all-encompassing method of meme analysis with a particular focus on the Telugu and Kannada languages.

Deep learning methodologies, notably Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), stand out for their prowess in automatically extracting features from meme images and captions, respectively. Transfer learning techniques have further propelled advancements in meme classification, with pre-trained models like BERT and GPT being fine-tuned on meme datasets to leverage their comprehensive understanding of textual and visual data. Multimodal approaches, which fuse text and image features, have garnered attention for their ability to capture the intricate interplay between different elements within memes.

There are various difficulties when it comes to multi-modal memes with text written in different languages. A notable obstacle concerns the incorporation and handling of heterogeneous linguistic material in the meme. Robust language recognition and processing techniques are required when handling text in various languages in order to correctly interpret and evaluate the textual components. Furthermore, it can be difficult to understand the context and feeling that the memes communicate because different languages have different grammatical nuances and cultural allusions. An additional problem is ensuring that the text and supporting graphic elements are aligned cohesively, which calls for advanced multi-modal fusion techniques. An additional layer of complexity is added by the need to guarantee the efficacy and accuracy of text extraction techniques across linguistic boundaries.

To address these challenges, we aim to accomplish the following objectives:

- Meme text extraction using OCR techniques and classifying domain and trolling.
- Generating Captions and combining it with extracted text for classification.
- Image classification with image transformer based models.

The remaining sections of paper is structured as follows: Section II gives a overview of past existing research. Section III describes the dataset used. In Section IV,proposed methodologies is mentioned. Section V describes our experiments and results. Finally, Section VI wraps up the paper and suggests future research directions.

II. Literature Survey

Manohar et.al in [2] presented a classification model designed to distinguish between troll memes and normal memes by analyzing both visual and textual elements. Leveraging a Convolutional Neural Network (CNN) for image classification, the model incorporates emotional values associated with memes to enhance accuracy. Text processing involves traditional natural language processing techniques, utilizing the Tesseract library for text extraction and Word2Vec modeling for feature extraction from meme text. The classification is based solely on displayed text, excluding memes without text. Results indicate a high accuracy rate of 98 percent, underscoring the effectiveness of the model in differentiating between troll and normal memes.

David et.al in [1] discusses a comprehensive approach to meme analysis, incorporating multiple modalities such as image, text, and facial expressions. It employs graph learning techniques to construct evolutionary trees that depict the development and variations of memes over time. The study focuses on a significant online discourse surrounding a political event, specifically the 2018 US mid-term elections, demonstrating the applicability of the proposed methodology in analyzing memes within specific contexts. Additionally, the paper outlines a technique for pre-processing meme images using Optical Character Recognition (OCR), which involves extracting text from memes to facilitate further analysis.

Conversely, Iyyer and Boyd-Graber (2022) presented a multimodal meme classification framework based on graph convolutional networks (GCNs). In order to capture the intricate interaction between textual and visual elements within memes, their model used graph-based representations, which allowed for more complex classification choices.

Kryven and Korshunova (2021) suggested cross-modal attention networks for multimodal meme classification, building on these methods. Their model dynamically weighted the contributions of textual and visual features by incorporating attention mechanisms, which improved classification performance and allowed for more efficient fusion of information across modalities.

Furthermore, Lee and Lee (2023) promoted the use of multitask learning in the classification of cross-modal memes. Their framework improved generalization and robustness across a variety of memes by simultaneously optimizing multiple related tasks, such as sentiment analysis and humor detection, alongside meme classification.

A novel approach to multimodal meme classification was proposed by Alsaedi et al. (2021) in an effort to overcome the difficulties caused by the combination of text and image modalities. They used cutting-edge machine learning strategies to produce precise classification outcomes. Comparably, Ding et al. (2020) investigated deep learning techniques for multimodal meme categorization, showcasing how well their method handled intricate meme material.

A graph convolutional network-based method for multimodal meme classification was presented by Iyyer and Boyd-Graber in 2022. This method improves classification accuracy by taking advantage of the relationships between meme components. Cross-modal attention networks for multimodal meme classification were the subject of Kryven and Korshunova's (2021) study, which stressed the significance of recording cross-modal interactions to improve classification performance.

Additionally, in order to enhance overall performance, Lee and Lee (2023) presented a multi-task learning framework for cross-modal meme classification, highlighting the advantages of collaboratively learning from several related tasks. All together, these research papers demonstrate the wide variety of approaches and strategies used in multimodal meme classification, highlighting ongoing efforts to push the boundaries of this field's state-of-the-art.

III. Dataset Description

For our classification task, we used the SCaLAR dataset, which consists of meme images in Telugu and Kannada, two languages spoken in South India. The three subsets of the dataset—training, testing, and validation—ensure reliable assessment and generalization of the created classification models. We extracted 2075 images out of which 1422, 305, and 205 images for training, testing, and validation, respectively, from the Telugu subset. A sample meme image is given in Fig.1 along with the extracted text and captions. In the same way, we obtained 2022 images out of which 1360, 301, and 341 images for the respective splits for the Kannada subset. Sorting images into three categories—politics, movies, and sports—and differentiating between 'troll' and 'not troll' images was the classification task. Our study sought to investigate the subtleties of meme classification in multilingual contexts by utilizing this diverse dataset, thereby contributing to a deeper understanding of online Social discourse and the distribution of content in South India. Word clouds for Kannada captions are mentioned in Fig.2 and for telugu in Fig.3.

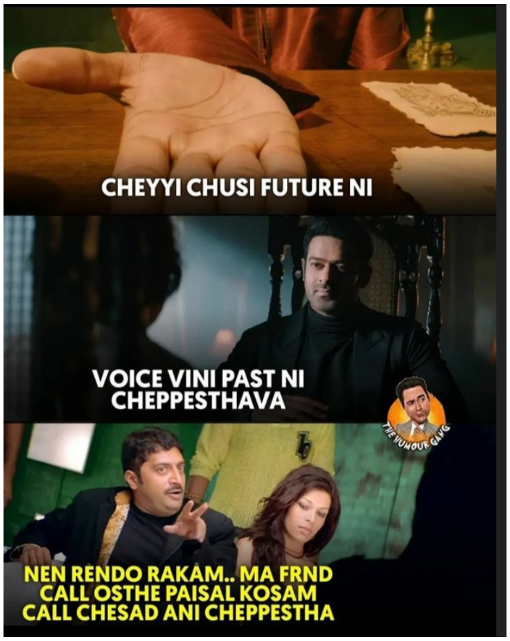

Fig. 1. Ssample Telugu Meme
Caption:Collage of photos a man in black suit
Extracted Text: Cheyyi chusi future ni voice vini past ni cheppesthava nen rendo rakam ma frnd call osthjhe paisal kosam calll chesad ani cheppestha

IV. Proposed Methodology

Our methodology involved several approaches: image classification to analyze meme visuals, extracted text classification for textual content, and combined text-caption classification for holistic understanding. We categorized memes into thematic domains like politics, sports or entertainment and distinguished between troll and non-troll memes using the respective models as shown in Fig.4.

A. *Meme Image classification*

1) OCR for Text extraction: Text extraction from Pyteseract images using OCR requires a few crucial steps. In order to

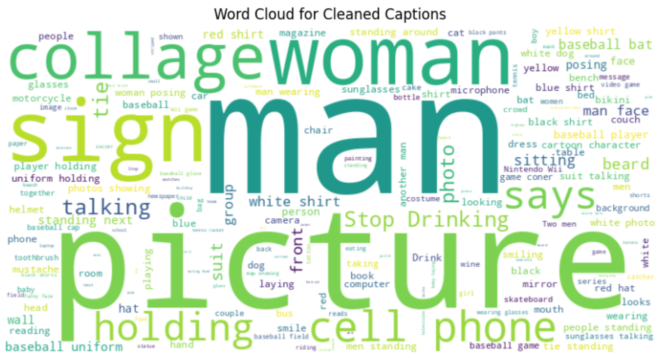

Fig. 2. Word Cloud for Kannada Captions

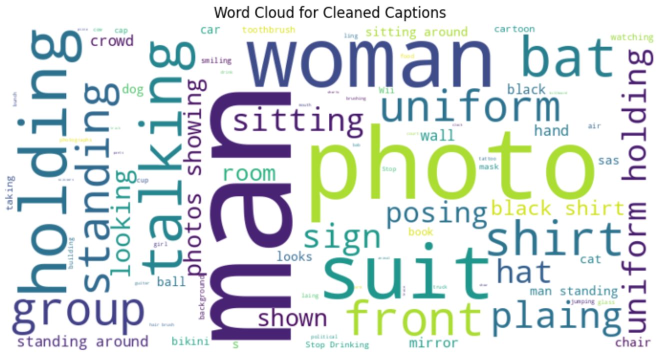

Fig. 3. Word Cloud for Telugu Captions

improve readability and eliminate any noise or artifacts that might impede the OCR process, the input image containing the text is first preprocessed. Preprocessing techniques like resizing, denoising, and binarization might be used. The Pyteseract OCR engine then receives the preprocessed image and uses optical character recognition algorithms to identify and extract text from the image. Pyteseract creates a textual representation of the image's content by analyzing the image and recognizing specific characters and words. Following extraction, the text can undergo additional processing or analysis as required for tasks like information retrieval or natural language processing.

2) Vision Transformers for Caption Generator: There are several essential steps in the Vision Transformer (ViT) based caption generation methodology. First, a Vision Transformer model processes the input images. It uses self-attention mechanisms to extract hierarchical features from the visual input. The complex visual semantics and contextual information contained in the images are captured by these features. The visual features that were extracted are then combined with token embeddings that represent the start-of-sequence token, which denotes the start of the caption generation process. After that, a transformer-based decoder architecture, such as a transformer decoder or a related sequence generation mode, receives this combined representation.

3) CNN based classification: Our designed CNN model architecture uses rectified linear unit (ReLU) activation functions to introduce non-linearity and consists of three convolutional layers followed by batch normalization and max-pooling layers. In order to reduce overfitting during training, we used the Adam optimizer in conjunction with the binary cross-entropy loss function and dropped out regularization after the completely connected layers. Standard metrics like accuracy were used to assess the model's performance after it had been trained for ten epochs.

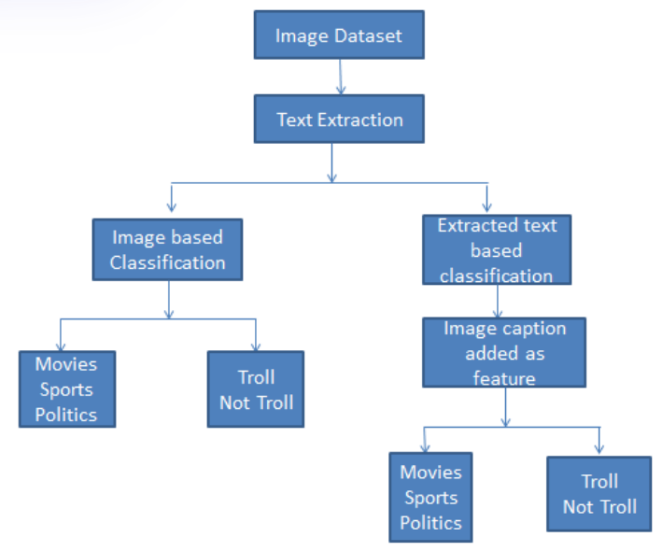

Fig. 4. Proposed Methodology

4) ResNet-50 based classification: A deeper convolutional neural network with the ResNet-50 architecture was employed for picture categorization. To overcome the vanishing gradient problem, this architecture comprises of numerous residual blocks with skip connections. To add non-linearity, rectified linear unit (ReLU) activation functions were applied within the residual blocks following every convolutional layer. The feature maps were downsampled using max-pooling layers and the normalization was done using batch normalization. To reduce overfitting, the training process used the Adam optimizer, dropout regularization after fully connected layers, and the binary cross-entropy loss function. After the model had been trained for ten epochs, its performance was assessed using common measures like accuracy.

5) Vision Transformers: The model is based on a pretrained Vision Transformer (ViT), a cutting-edge architecture designed for image categorization applications.To enable quick model training, first, necessary modules are imported, such as the engine module from the going_modular.going_modular package. The pretrained ViT model's parameters are the precise target of an Adam optimizer that is instantiated with a learning rate of 1e-3. Simultaneously, a Cross Entropy Loss function is established to calculate the loss throughout the training phase, enabling the model's optimization. The training process is then coordinated by invoking the engine.train function, which iteratively optimizes the model parameters over 10 epochs to guarantee efficient convergence of the model's performance. Through the training and test dataloaders that are provided, this function effectively obtains the necessary data to allow for a thorough assessment of the model's performance.

B. Extracted Text Classification

1) Machine Learning models for classification: Memes were classified using machine learning models such as Random Forest, XGBoost, Naive Bayes, SVM (Support Vector Machine), and Logistic Regression by utilizing TF-IDF embeddings. TF-IDF embeddings were used to represent the text content that was extracted using Pytesseract, which made the classification process easier. By using this method, the models were able to accurately represent the semantic meaning that was embedded in the textual material, which made it possible to classify the content using learnt representations.

2) Bert Model: There are a few key steps in the process of developing a BERT model for meme classification. Usually, a pre-trained BERT variant, like BERT-base or BERT-large, is used to instantiate the BERT model at first. The text is then tokenized and transformed into input features like input word IDs, input masks, and segment IDs in order to comply with the tokenization requirements of the BERT model. The BERT model receives these input features and uses them to create contextualized embeddings for each token in the input text. Additional layers are applied to the embeddings to further process them; these layers frequently include dense layers to extract higher-level features and dropout layers to prevent overfitting.Lastly, the model's output layer is set up according to the classification task; for example, a sigmoid activation function is usually used for binary classification, and a softmax activation function is usually used for multi-class classification.

3) RoBERTa Model: There are several crucial steps in the model-building process. Tokenization and attention mechanisms depend on the inputs for the RoBERTa model, which we first initialize. These inputs include input word IDs, input masks, and input type IDs. After that, we import the RoBERTa model, which has been pre-trained on a sizable corpus of data, from the HuggingFace library. We obtain contextual embeddings for every token by running the input sequences through the RoBERTa model. We extract the embeddings as the first position output since Huggingface transformers generate multiple outputs. Next, in order to flatten the output tensor and avoid overfitting, we apply a dropout layer. In order to extract high-level features, we then feed the flattened tensor into a dense layer with 256 units and ReLU activation. Lastly, we join the dense layer to an output layer that predicts the probability distribution among the categories using softmax activation. Using a learning rate of 1e-5, the model is compiled using the Adam optimizer, and the model parameter is optimized using sparse categorical cross-entropy loss.

C. Caption+Extracted Text Classification

1) ML models: Memes were classified using machine learning models such as Random Forest, SVM (Support Vector Machine) by utilizing TF-IDF embeddings of both caption and extracted text. TF-IDF embeddings were used to represent the text content that was extracted using Pytesseract module and then preprocessed by removing unwanted text, numbers, symbols. By using this method, the models were able to accurately represent the semantic meaning that was embedded in the textual material, which made it possible to classify the content using learnt representations.

2) Custom ANN model: Caption Text and extracted text were combined and classified using a neural network model based on TF-IDF vectors. To make neural network processing easier, the TF-IDF vectors were first transformed from sparse matrices to dense arrays. An input layer, two hidden layers with 64 and 32 neurons each, and an output layer with a single neuron that used sigmoid activation for binary classification made up the neural network architecture. The binary cross-entropy loss function and Adam optimizer were used to create the model, and accuracy was used as the evaluation metric. Fitting the model to the training data for 15 epochs with a batch size of 32 and validating it on a different validation dataset comprised the training phase. Ultimately, an independent test set was used to evaluate the model's performance using metrics like accuracy and loss.

3) Multilingual Bert(mBert) Model: We used a deep learning approach based on the multilingual BERT (Bidirectional Encoder Representations from Transformers) model for sequence classification to tackle the multimodal meme classification task. First, we preprocessed the meme text by truncating or padding sequences up to a maximum of 128 tokens using the BERT tokenizer. After being preprocessed, the text data was tokenized and turned into input IDs so that the BERT model could use it. We then built data loaders and PyTorch datasets to handle the input data during training more effectively. The BERT model, which was adjusted for our particular task, was the model architecture selected for sequence classification.The BERT model, which was adjusted for our particular task, was the model architecture selected for sequence classification. Two output labels were assumed for binary classification. We employed a learning rate scheduler to dynamically modify the learning rate throughout training epochs and updated the model parameters using the AdamW optimizer. Using batch iteration through the training dataset and backpropagation to update the model parameters in order to minimize loss, we trained the model for five epochs. We tracked the model's performance on the validation set during training, evaluating the model's convergence and generalization by computing the validation loss. In addition, we assessed the model's accuracy by calculating the test loss and producing classification reports, as well as by evaluating the model's performance on the test set following training completion.

4) XLM Roberta Model: Our method for multimodal meme classification made use of the RoBERTa model's pre-trained on a variety of languages XLM-RoBERTa (Cross-lingual Language Model - RoBERTa) variant. By using XLM-RoBERTa, multilingual meme content can be handled more efficiently, allowing for reliable classification in a variety of linguistic contexts. To ensure consistent representation across various languages, we first preprocessed the textual content of memes using the XLM-RoBERTa tokenizer. To help with model processing, the textual data was tokenized, transformed into input IDs, and sequences were padded or truncated up to a maximum of 128 tokens. We used the XLM-RoBERTa model, which was optimized for our particular task, for sequence classification. Assuming a multiclass classification scenario, we set the number of output labels to three. To improve model convergence and stability, we developed a learning rate scheduler and used the AdamW optimizer to update the model parameters. This allowed us to dynamically modify the learning rate during training epochs.

5) CLIP Model: In order to clarify the intricate interactions between textual and visual data, we utilized the CLIP (Contrastive Language-Image Pre-training) model. By utilizing its creative architecture, CLIP combines two backbones: one for handling textual data and the other for processing image data. The vision backbone uses a version of the Vision Transformer (ViT) architecture and instead of using traditional convolutional layers, it uses transformer-inspired self-attention mechanisms to extract hierarchical features from images. Meanwhile, the language backbone functions on textual inputs, producing embeddings that capture the semantic meaning of textual descriptions, akin to models such as BERT. A significant breakthrough in AI architecture, the integration of dual backbones enables a comprehensive comprehension of multimodal data. By means of well-crafted projection heads and normalization layers. integrating elements from both modalities into a common semantic space to promote significant connections between text and images. Through the use of a contrastive learning objective in training, CLIP is able to distinguish between semantically related pairs—that is, pairs of images and the text descriptions that go with them—and unrelated pairs, thus enabling it to navigate the complex relationship that exists between images and text. Initially, features are extracted from the provided dataset—which consists of pictures and captions—using the CLIP model. For this, a function is used. Every image-caption pair is iteratively processed within this function. The model is set up to operate on the right kind of computing device (CPU or GPU, for example). The Image.open function is used for each pair to load the image, and then the image and caption are transformed into input tensors that meet the specifications of the model. After that, these inputs are sent to the specified computing device. Using the supplied inputs, the model calculates the image-text similarity scores, also known as logits per image. After that, these scores are transformed into numpy arrays and added to a feature list. After doing this for each image-caption pair in the dataset, a set of features representing the images in the dataset are produced. The features are further processed for classification tasks after they are extracted. The extracted features in this study are used to train a classifier, such as logistic regression. To make the features compatible with the classifier, they are reshaped as necessary. The classifier is tested on both the validation and test datasets after being trained using the features that were taken from the training dataset. Using these datasets, predictions are generated by the trained classifier. Precision, recall, F1-score, accuracy, and other performance metrics are calculated based on the classifier's predictions. These metrics give information about how well the classifier has classified instances from the test and validation datasets.

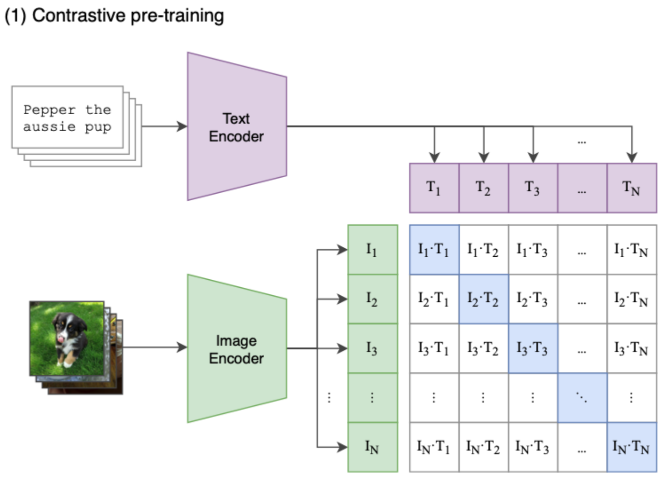
(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
T_1 T_2 T_3 ... T_N
I_1 I_2 I_3 ⋮ I_N
$I_1{\cdot}T_1$ $I_1{\cdot}T_2$ $I_1{\cdot}T_3$ ... $I_1{\cdot}T_N$
$I_2{\cdot}T_1$ $I_2{\cdot}T_2$ $I_2{\cdot}T_3$ ... $I_2{\cdot}T_N$
$I_3{\cdot}T_1$ $I_3{\cdot}T_2$ $I_3{\cdot}T_3$ ... $I_3{\cdot}T_N$
$I_N{\cdot}T_1$ $I_N{\cdot}T_2$ $I_N{\cdot}T_3$ ... $I_N{\cdot}T_N$

Fig. 5. CLIP feature Extraction

V. Experiments & Results

A. Meme Image classification

Telugu Language		
Model Name	Validation Accuracy	Test Accuracy
CNN	62.48	58.49
ResNet-50	56.20	50.98
Image Transformers	70.64	53.94
Kannada Language		
Model Name	Validation Accuracy	Test Accuracy
CNN	60.80	52.68
ResNet-50	56.38	51.98
Image Transformers	72.89	55.68

TABLE I
MEME IMAGE CLASSIFICATION DOMAIN

Telugu Language		
Model Name	Validation Accuracy	Test Accuracy
CNN	60.56	54.44
ResNet-50	59.78	53.89
Image Transformers	68.54	57.98
Kannada Language		
Model Name	Validation Accuracy	Test Accuracy
CNN	59.52	58.97
ResNet-50	60.78	58.98
Image Transformers	73.87	59.60

TABLE II
MEME IMAGE CLASSIFICATION TROLL/NON TROLL

B. Extracted Text Classification

C. Caption+Extracted Text Classification

The evaluation findings for the Telugu domain categories' multiclass classification and binary classification of troll content that came from the CLIP model. The model demonstrated its ability to accurately classify instances as troll content,

-	Precision	F1 Score	Accuracy
Validation Troll	0.75	0.74	0.73
Test Troll	0.74	0.75	0.75
Validation domain	0.65	0.64	0.67
Test domain	0.68	0.64	0.64

TABLE III
Telugu Validation and Test Metrics

as evidenced by its precision of 0.75 and F1-score of 0.74 on the validation dataset for the binary classification task of identifying troll content. Similar to this, the model performed consistently on the test dataset, obtaining an F1-score of 0.75 and a precision of 0.74 for troll classification. These findings highlight the model's dependability and robustness in identifying trolling content from legitimate content across a variety of datasets. The CLIP model performed admirably in the multiclass classification task designed to determine domain categories in Telugu content. The model's precision on the validation dataset was 0.65, demonstrating its ability to correctly class.

-	Precision	F1 Score	Accuracy
Validation Troll	0.81	0.80	0.81
Test Troll	0.81	0.83	0.82
Validation domain	0.69	0.70	0.73
Test domain	0.70	0.68	0.74

TABLE IV
Kannada Validation and Test Metrics

For Telugu and Kannada languages, the CLIP model performed well in both binary troll content classification and multiclass domain categorization tasks. In Telugu, the model consistently produced a precision of 0.74 and an F1-score of 0.75 on the test dataset, while achieving a precision of 0.75 and an F1-score of 0.74 on the validation dataset for troll content identification. It achieved a precision of 0.65 and an F1-score of 0.64 in domain categorization on validation; on test, it improved to a precision of 0.68 while keeping the same F1-score. With precision, recall, and an F1-score of 0.81 for troll classification on both test and validation sets, the model performed even better for Kannada.It obtained an accuracy of 0.69 in domain categorization during validation, rising to 0.70 during testing, with corresponding F1-scores of 0.70 and 0.68. These findings highlight the model's resilience and efficiency in Telugu and Kannada language classification of troll content and domain categories. Perceptive outcomes were obtained from evaluating different machine learning models for multimodal meme classification in Telugu and Kannada. Logistic regression, naive Bayes, random forest, support vector machine (SVM), XGBoost, RoBERTa, BERT, and multilingual BERT (m-BERT) models were tested in both languages. With validation and test accuracies of 72.86% and 70.90% in Telugu, respectively, m-BERT outperformed other models like logistic regression, naive Bayes, and SVM. On the other hand, the RoBERTa model performed relatively worse, with test and validation accuracies of 55.60% and 57.60%, respectively. m-BERT also proved to be the best model in the Kannada language, with test and validation accuracies of 73.59% and 75.68%, respectively. Significantly, BERT performed well in both languages, with validation accuracy of 64.38% in Telugu and 70.26% in Kannada. Apart from the multimodal meme classification, the BERT model's text classification of troll and non-troll content produced encouraging results, though no precise accuracy metrics were given. Overall, these results highlight the BERT and m-BERT models' effectiveness for multimodal meme classification tasks across linguistic contexts and highlight how far these models can be taken to improve meme analysis and comprehension in online communication platforms.

Telugu Language		
Model Name	Validation Accuracy	Test Accuracy
Logistic regression	60.32	56.64
naive bayes	59.64	57.29
svm	60.68	56.26
random forest	58.48	56.25
xg boost	56.75	55.25
Roberta	54.10	58.60
Bert	60.30	59.80
m-Bert	68.84	66.90
Kannada Language		
Model Name	Validation Accuracy	Test Accuracy
Logistic regression	62.58	59.84
naive bayes	63.50	59.71
svm	60.00	56.72
random forest	61.59	60.46
xg boost	64.56	60.59
Roberta	65.72	63.50
Bert	68.72	67.50
m-Bert	74.50	72.50

TABLE V
Extracted Text classification Troll/Non troll

Telugu Language		
Model Name	Validation Accuracy	Test Accuracy
Logistic regression	60.26	58.68
naive bayes	61.87	59.68
svm	59.68	57.26
random forest	62.3	60.12
xg boost	58.26	56.80
Roberta	55.20	57.60
Bert	64.38	62.89
m-Bert	72.86	70.90
Kannada Language		
Model Name	Validation Accuracy	Test Accuracy
Logistic regression	64.84	60.29
naive bayes	62.48	60.71
svm	63.28	61.38
random forest	66.89	63.59
xg boost	62.40	59.68
Roberta	65.89	62.84
Bert	70.26	68.29
m-Bert	75.68	73.59

TABLE VI
Extracted Text classification based on Domain

Promising outcomes were observed in the performance evaluation of different machine learning models for multimodal meme classification in Telugu and Kannada. Multilingual BERT (m-BERT) was the best-performing model in both languages, with the highest test and validation accuracies. To be more precise, m-BERT obtained 72.86% and 70.90% validation and test accuracies in Telugu and 75.68% and

73.59% in Kannada, respectively. With validation and test accuracies of 64.38% and 62.89% in Telugu and 70.26% and 68.29% in Kannada, respectively, BERT demonstrated competitive performance as well. On the other hand, models such as RoBERTa showed comparatively poorer accuracy. These results highlight how well m-BERT and BERT models perform in Telugu and Kannada multimodal meme classification tasks.

Telugu Language		
Model Name	Validation Accuracy	Test Accuracy
SVM	68.80	62.68
Random Forest	70.25	64.60
Custom ANN	73.80	72.62
mBERT	76.80	74.30
xlm-Roberta	80.46	76.54
Kannada Language		
Model Name	Validation Accuracy	Test Accuracy
SVM	66.70	64.80
Random Forest	68.80	67.50
Custom ANN	72.80	70.40
mBERT	74.5	70.12
xlm-Roberta	70.38	68.80

TABLE VII
Captions + Extracted Text classification based on Troll/No troll

Considerable differences in performance were found when evaluating machine learning models for multimodal meme classification in Telugu and Kannada. With the highest validation and test accuracy in Telugu, 80.46% and 76.54%, respectively, for the xlm-RoBERTa model, and 74.30% and 76.80%, respectively, for mBERT, were the next closest models. Notably, a specially designed artificial neural network (ANN) also showed impressive results, achieving 72.62% test accuracy and 73.80% validation accuracy. On the other hand, mBERT performed significantly better in Kannada than other models, such as SVM and Random Forest, with a validation accuracy of 62.3% and a test accuracy of 60.12%. These results highlight the potential for reliable and accurate meme analysis of the xlm-RoBERTa and mBERT models for multimodal meme classification, especially in Telugu.

Telugu Language		
Model Name	Validation Accuracy	Test Accuracy
SVM	66.45	60.80
Random Forest	69.50	63.80
Custom ANN	70.60	70.15
mBERT	76.80	71.20
xlm-Roberta	71.56	70.25
Kannada Language		
Model Name	Validation Accuracy	Test Accuracy
SVM	68.50	66.20
Random Forest	64.60	62.50
Custom ANN	70.50	68.26
mBERT	78.20	72.40
xlm-Roberta	72.6	68.25

TABLE VIII
Captions + Extracted Text classification based on Domain

VI. Conclusion & Future scope

The study of machine learning models for multimodal meme classification in Telugu and Kannada has yielded important information about the effectiveness of different strategies. The assessment showed significant differences in the two languages' model performances, indicating the impact of linguistic subtleties on classification accuracy. Although certain models showed encouraging outcomes in one language, their efficacy decreased in the other, underscoring the significance of taking language-specific traits into account when developing models. These results highlight the difficulty of classifying multimodal memes and the demand for reliable, language-neutral methods. Furthermore, the research highlights the importance of transformer-based models in managing various linguistic environments, demonstrating their capacity to promote multimodal meme analysis in various languages. Moreover, the study offers significant implications for future research, including investigating new architectures and optimizing training techniques to enhance classification precision. All things considered, this work establishes the foundation for future developments in the field of multimodal meme classification and adds to our understanding of it.

Future research in multimodal meme classification in Telugu and Kannada languages should explore language-specific model optimization, leveraging advanced models like Navarasa Indian Languages Model and Indigo NLP Model. Fine-tuning machine learning models on diverse datasets tailored for these languages can enhance classification accuracy by capturing linguistic nuances and cultural slangs. Expanding the multimodal dataset to include a wider range of memes, spanning variations in language, humor, and cultural references, could improve the model's comprehension and classification abilities. Integrating state-of-the-art models offers potential insights for effectively analyzing multimodal memes across different languages and cultural contexts.

REFERENCES

[1] Beskow, David M., Sumeet Kumar, and Kathleen M. Carley. ”The evolution of political memes: Detecting and characterizing internet memes with multi-modal deep learning.” Information Processing & Management 57, no. 2 (2020): 102170.

[2] Shridara, Manohar Gowdru, Daniel Hládek, Matúš Pleva, and Renát Haluška. ”Identification of Trolling in Memes Using Convolutional Neural Networks.” In 2023 33rd International Conference Radioelektronika (RADIOELEKTRONIKA), pp. 1-6. IEEE, 2023.

[3] Hossain, Eftekhar, Omar Sharif, and Mohammed Moshiul Hoque. ”NLP-CUET@ DravidianLangTech-EACL2021: investigating visual and textual features to identify trolls from multimodal social media memes.” In Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pp. 300-306. 2021.

[4] Aslam, Nida, Irfan Ullah Khan, Teaf I. Albahussain, Nouf F. Almousa, Mizna O. Alolayan, Sara A. Almousa, and Modhi E. Alwhebi. ”MEDeep: A Deep Learning Based Model for Memotion Analysis.” Mathematical Modelling of Engineering Problems 9, no. 2 (2022).

[5] Premjith, B., Bharathi Raja Chakravarthi, Malliga Subramanian, B. Bharathi, Soman Kp, V. Dhanalakshmi, K. Sreelakshmi, Arunaggiri Pandian, and Prasanna Kumaresan. ”Findings of the shared task on multimodal sentiment analysis and troll meme classification in Dravidian languages.” In Proceedings of the second workshop on speech and language technologies for Dravidian languages, pp. 254-260. 2022.

[6] . Smitha, E. S., Selvaraju Sendhilkumar, and G. S. Mahalaksmi. ”Meme classification using textual and visual features.” In Computational Vision and Bio Inspired Computing, pp. 1015-1031. Springer International Publishing, 2018.

[7] .Dai, Wenhao, Xing Fang, Haoyi Zhou, Jingwen Hu, Xiaoyong Du, and Jianping Yin. ”MMCNet: A Multi-Modal Fusion Model for Meme Classification.” In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pp. 7002-7013. 2021.

[8] . Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. ”XLM-R: Combining Multilingual Pretrained Models for Meme Classification.” In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pp. 6880-6890. 2021.

[9] Singh, Akash Kumar, Amarjeet Singh, and Ganesh Bagler. ”M3EM: A Multimodal Meme Embedding Method for Meme Classification.” In 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1-8. IEEE, 2021.

[10] Goyal, Yash, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. ”ACT: Adaptive Computation Time for Recurrent Neural Networks.” In Advances in Neural Information Processing Systems, pp. 1406-1416. 2017.

[11] Karmakar, Sukarna, Chen Wu, Peixiang Zhong, Min Zhang, and Jianming Wei. ”MemeNet: A Deep Multimodal Fusion Model for Meme Classification.” In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 4730-4739. 2020.

[12] Huang, Qingcai, Qian Peng, Zhonghao Sheng, and Xuanjing Huang. ”Leveraging Multimodal Transformer for Meme Classification.” In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 15, pp. 13361-13368. 2021.