IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 December 2023, accepted 17 January 2024, date of publication 1 February 2024, date of current version 15 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3361322

RESEARCH ARTICLE

Multimodal Hate Speech Detection in Memes Using Contrastive Language-Image Pre-Training

GREESHMA ARYA[1], MOHAMMAD KAMRUL HASAN[2], (Senior Member, IEEE), ASHISH BAGWARI[3], (Senior Member, IEEE), NURHIZAM SAFIE[2], (Member, IEEE), SHAYLA ISLAM[4], (Senior Member, IEEE), FATIMA RAYAN AWAD AHMED[5], AAISHANI DE[1], MUHAMMAD ATTIQUE KHAN[6,7], (Senior Member, IEEE), AND TAHER M. GHAZAL[8,9,10], (Senior Member, IEEE)

[1]Department of Electronics and Communication Engineering, Indira Gandhi Delhi Technical University for Women, New Delhi 110006, India
[2]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), Bangi, Selangor 43600, Malaysia
[3]Department of Electronics and Communication Engineering, Uttarakhand Technical University, Dehradun 248007, India
[4]Institute of Computer Science and Digital Innovation, UCSI University Malaysia, Kuala Lumpur 56000, Malaysia
[5]Computer Science Department, College of Computer Engineering and Science, Prince Sattam Bin Abdulaziz University, Al-Kharj 16273, Saudi Arabia
[6]Department of Computer Science, HITEC University, Taxila 47080, Pakistan
[7]Department of CS and Mathematics, Lebanese American University, Beirut 1102 2801, Lebanon
[8]Centre for Cyber Physical Systems, Computer Science Department, Khalifa University, Abu Dhabi, United Arab Emirates
[9]Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), Bangi, Selangor 43600, Malaysia
[10]Applied Science Research Center, Applied Science Private University, Amman 11937, Jordan

Corresponding authors: Mohammad Kamrul Hasan (hasankamrul@ieee.org), Nurhizam Safie (nurhizam@ukm.edu.my), Ashish Bagwari (ashishbagwari@ieee.org), and Shayla Islam (shayla@ucsiuniversity.edu.my)

This work was supported in part by the Universiti Kebangsaan Malaysia under Grant GUP-2023-010.

ABSTRACT In contemporary society, the proliferation of online hateful messages has emerged as a pressing concern, inflicting deleterious consequences on both societal fabric and individual well-being. The automatic detection of such malevolent content online using models designed to recognize it, holds promise in mitigating its harmful impact. However, the advent of ''Hateful Memes'' poses fresh challenges to the detection paradigm, particularly within the realm of deep learning models. These memes, constituting of a textual element associated with an image are individually innocuous but their combination causes a detrimental effect. Consequently, entities responsible for disseminating information via web browsers are compelled to institute mechanisms that regulate and automatically filter out such injurious content. Effectively identifying hateful memes demands algorithms and models endowed with robust vision and language fusion capabilities, capable of reasoning across diverse modalities. This research introduces a novel approach by leveraging the multimodal Contrastive Language-Image Pre-Training (CLIP) model, fine-tuned through the incorporation of prompt engineering. This innovative methodology achieves a commendable accuracy of 87.42%. Comprehensive metrics such as loss, AUROC, and f1 score are also meticulously computed, corroborating the efficacy of the proposed strategy. Our findings suggest that this approach presents an efficient means to regulate the dissemination of hate speech in the form of viral meme content across social networking platforms, thereby contributing to a safer online environment.

INDEX TERMS CLIP, facebook hateful meme dataset, multimodal, contrastive learning, zero-shot prediction, InfoNCE contrastive loss, prompt engineering, cosine similarity matrix.

I. INTRODUCTION

The pervasive use of hateful memes as a vehicle for spreading animosity on online platforms has become an alarming trend. The term ''hate speech'' has solidified its presence as a ubiquitous phenomenon in the realm of the internet. Memes, which can be any shareable content encompassing photographs or videos that are spread, altered, and repeated over time [1], serve as conduits for the transmission of such hate among individuals. A speech exhibiting hostility or violence

The associate editor coordinating the review of this manuscript and approving it for publication was Tai Fei.

towards individuals who have protected traits safeguarded by societal norms or constitutional provisions is categorized as hateful speech. The direct and automatic identification of hateful memes assumes paramount significance in fostering a healthy social networking environment. Proactively filtering out content that disseminates hatred before it reaches online platforms plays a pivotal role in curbing the proliferation of social media hate.

Hateful memes are characterized by a blend of background images and text captions, which encapsulate the intentions and sentiments of users. While these images or captions may appear harmless when viewed in isolation, their combination elicits a disturbing effect. This hidden meaning in the combination is very much visible to the human mind but is often concealed from conventional scanners as it is challenging for machines to recognize this difference. As a result, hateful users intentionally post hate speech in the form of such offensive memes to get past the traditional scanners. Traditional automatic text detection techniques and visual feature analysis work in isolation totally disregarding visual and text features respectively, rendering it challenging to identify multimodal hate speech. To address this, a model must transcend the limitations of processing individual modalities to combine the processing of two or more modalities to comprehend the complexities inherent in the amalgamation of text and images.

Consider phrases such as ''look how many people love you'' or ''you look beautiful today.'' Paired with seemingly innocuous images of a skunk or tumbleweed, these otherwise benign statements transform into destructive and mean-spirited expressions. This perceptual nuance, easily discerned by humans, poses a formidable challenge for AI systems. Despite recent initiatives to detect hate speech in text and images, there is a notable scarcity of models adept at recognizing hate speech in multimodal contexts that encompass both text and images.

This study draws upon the ''Facebook Hate Meme Dataset,'' a rich repository of over 10,000 freshly curated multimodal examples (text + image) generated by Facebook AI, as its primary data source. The proposed methodology employs the CLIP (Contrastive Language-Image Pre-training) model to analyze the accompanying text and images, to determine whether their combined expression qualifies as hateful. Beyond mere detection, the approach holds the potential to proactively filter out harmful memes, contributing to the overarching goal of mitigating the spread of hate on social media platforms.

The implementation process involves training the model on the dataset developed by Facebook and subsequently subjecting it to rigorous testing using a diverse collection of images, with the model's accuracy serving as a quantitative benchmark. Notably, the CLIP model exhibits high accuracy in identifying hostile memes, marking a significant stride in the ongoing efforts to fortify the digital landscape against the deleterious impact of hate speech online.

A. MOTIVATION

The rise of online hateful messages in recent times has emerged as a pressing societal concern, posing significant harm to both individuals and the community at large [2]. These ''*Hateful Memes*'' represent a particularly vexing challenge. The motivation behind addressing this issue lies in the imperative to curb the adverse consequences of such content. Following the Covid-19 pandemic, social media has exploded, and hate speech in the form of memes is being widely used adversely for cyberbullying as well as discrimination against minority communities in a variety of ways, including racism, sexism, sexual harassment, and discrimination based on one's sexual orientation or religious background. Hence, it becomes vital to regulate such hostile content to safeguard users as well as establish clean and secure social networking platforms [3], [4]. The unique nature of hateful memes, where seemingly innocuous text combines with images to create harmful effects, underscores the need for automated detection mechanisms. It becomes evident that internet platforms and companies responsible for delivering content to users must play an active role in filtering out these harmful messages. To accomplish this, the key lies in developing algorithms and models that exhibit robust fusion of vision and language capabilities, allowing them to reason effectively across diverse modalities that would be contributing to the mitigation of online hate and its far-reaching consequences.

B. CONTRIBUTION

This research makes a substantial contribution by addressing the pressing issue of online hate through the development and application of advanced deep learning models. The crux of the novelty in this research resides in the introduction and execution of a pioneering methodology that revolves around the integration of the multimodal CLIP model combined with a strategic application of prompt engineering. This fusion forms the basis of a unique approach to the classification of hateful and non-hateful memes. The technique capitalizes on the cutting-edge capabilities of the CLIP model, by incorporating which the study aims to harness trhe synergistic relationship between language and image features, enhancing the overall discriminative power of the proposed meme classification system. Central to this innovation is the recognition of the pivotal role played by accurate textual prompts, meticulously designed to serve as explicit classes for accurate and nuanced meme categorization. The precision and relevance of these prompts are systematically crafted to counter challenges such as false positives, thereby significantly elevating the discriminatory accuracy of the proposed method. Thus, this paper delves into the intricacies of this novel approach of synergy between the multimodal model and prompt engineering, which facilitates enhanced accuracy and efficacy in meme classification. Through the proposed methodology, an accuracy rate of 87.42% is attained in the detection of hateful memes. These

empirical results demonstrate a noteworthy improvement in classification accuracy as compared to other state-of-the-art models, which is directly attributed to the integration of accurate prompts into the CLIP model. Furthermore, this research extends its contribution to the broader context of online content regulation. The insights gained from this work can inform the development of content filtering systems by social networking companies and platforms, ultimately creating safer online environments. By offering a robust solution that effectively addresses the fusion of text and images in hateful content, this research takes a meaningful step towards mitigating the societal impacts of online hate speech and promoting a healthier digital ecosystem for all users.

The remaining sections are organized as follows: Section II discusses a few recent works related to hate speech detection in multimodal memes using various techniques and their learning. Then the problem definition, proposed optimal solution and motivation behind the selection are discussed in section III. It also contains the detailed description of the proposed framework including its working, architecture, and technical procedure. Then, the results and predicted outcomes obtained using the proposed model is deliberated in section IV. Finally, section V contains the future scope of the method and section VI concludes the project.

II. LITERATURE SURVEY

The field of hate speech identification has seen a lot of effort, but relatively little of it has focused on multimodal hate speech detection. Both NLP and network science have studied hate speech in depth. Hate speech has long been detected using language analysis. One of the key places where hate speech is targeted with a range of targets is social media. Obtaining a sentence embedding and putting it into a binary classifier for hate speech prediction are the typical processes for hate speech detection. A variety of language-based hate speech identification datasets have been made available for research on hate speech detection. According to ***Yang*** et al. adding image embedding information to text improves hate speech recognition ability right away. With the aid of Crowdflower employees, ***Hosseinmardi*** et al. categorize a dataset of Instagram photographs and the comments that go with them. Two inquiries were made to the staff: First, does the case represent cyberaggression, and second, does it represent cyberbullying. They demonstrate that incorporating picture characteristics enhances classification efficiency. The dataset included 998 cases, 90% of which had high confidence ratings, and 52% of which were labeled as bullying [5]. ***Zhong*** et al. compiled a dataset of 3000 samples of Instagram posts and comments in a manner like this. Two employees of Mechanical Turk were questioned: Is there any bullying in the comments? If so, can the bullying be linked to the image's subject matter? 560 incidents of bullying were discovered. They evaluate several features and simple classifiers for automatically identifying bullying [6].

A triplet is created by stacking the visual features, object tags, and text features of memes produced by the object detection model known as Visual features in Vision-Language (VinVl) and the optical character recognition (OCR) technology in the study by ***Yuyang Chen1, Feng PanID2, "Multimodal detection of hateful memes by applying a vision-language pre-training model"*** to perform cross-modal meme learning. After being tweaked and coupled with a random forest (RF) classifier, our model (OSCAR+RF) outperformed the other eleven (11) published baselines on the task of identifying nasty memes, reaching average accuracy and AUROC of 0.684 and 0.768, respectively, in a public test set. In conclusion, our study has demonstrated that VL-PTMs with anchor point additions can improve the efficiency of deep learning-based hate meme identification by including a more robust deep learning model [7].

The study by ***Yi Zhou1, Zhenhao Chen2, and Huiyuan Yang on MULTIMODAL LEARNING FOR HATEFUL MEMES*** identification focuses on the identification of hate speech using a model based on Visual Question Answering [8].

The research on ***"The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes" by Douwe Kiela, HamedFirooz, Aravind Mohan, VedanujGoswami, Amanpreet Singh, Pratik Ringshia, and DavideTestuggine*** suggests a new challenge set for multimodal classification that focuses on spotting offensive language in multimodal memes [9]. Problematic instances are included in the dataset to make it harder to rely on unimodal models and to demonstrate the superiority of multimodal models signals. Despite requiring complex reasoning, the task can be quickly reduced to a binary classification problem.

III. METHODOLOGY

The detection of Hateful Memes represents a binary classification challenge, seeking to ascertain whether a meme is offensive or hateful through the analysis of multimodal data constituting both text and image signals. Given the inherent complexity of memes, characterized by the coexistence of two modalities, and recognizing the formidable obstacle of detection accuracy in this task, a multi-task learning technique is deemed essential for drawing meaningful statistical conclusions [10], [11].

In this study, the categorization of multimodal hostile memes is characterized as a classification model that predicts the label of a multimodal meme (hateful or non-hateful) based on the associated image and text. To achieve this, models must predict a probability vector $y \in R$ over the two classes. In greater detail, y0 denotes the projected probability that the meme is non-hateful, while y1 represents the probability that the meme is hateful. If $y1 > y0$, the meme is classified as hateful; otherwise, it is categorized as non-hateful. Leveraging the Contrastive Language-Image Pre-training (CLIP) model, we successfully classify memes within this framework, assigning them to the hateful or

non-hateful classes based on the highest cosine similarity score in contrastive learning.

The technique employs a pre-trained model for feature extraction from a meme in a transfer learning context, combined with a downstream classification model that utilizes these features [12]. For operating without the addition of new data or manual labeling, a model that can balance the unambiguous information from the multimodal and the fuzzy information from the individual modality while minimizing generalization blunders is required. Consequently, a strategy that integrates statistical theory with state-of-the-art neural networks and optimization methods was developed to discern the offensive memes.

CLIP is one of the most significant advancements in computer vision, as it bridges the domains of computer vision and natural language processing. The challenges of huge datasets and subpar real-world results in conventional vision models are also addressed by the CLIP network. It distinguishes itself by allowing a singular method, CLIP, to handle a diverse range of applications without necessitating the construction of extensive custom datasets [13]. This departure from the conventional approaches, such as training models like ResNet requiring vast labeled image datasets, positions CLIP as a widely preferred model in the field of computer vision.

A. CONTRASTIVE LANGUAGE–IMAGE PRE-TRAINING

CLIP is a robust and scalable state of the art multi modal vision and language model introduced by OpenAI. This neural network boasts versatility, as it can be applied to various visual classification benchmarks and adeptly learns visual concepts through natural language supervision. Unlike traditional models, CLIP exhibits remarkable ''zero-shot'' capabilities akin to GPT-2 and GPT-3, enabling it to perform tasks such as predicting the most pertinent text snippet given the names of the visual categories to be recognized (image), without direct optimization.

Central to CLIP's prowess is its training methodology employing contrastive learning, which aims to map images and text descriptions into a shared latent space. This unique approach enables CLIP to discern whether an image and textual description match, therefore, facilitating tasks like image classification through text-image similarity [14], [15]. This means that CLIP can successfully predict which captions correspond to which images without domain-specific training, making it particularly potent for out-of-the-box text and image search applications [16]. Beyond its fundamental capabilities, CLIP finds application in a myriad of domains, including image generation, image similarity search, image ranking, object tracking, robotics control, image captioning, geo-localization, and more.

Its versatility arises from its comprehensive understanding of the intricate relationships between visual data and the corresponding linguistic representations. This profound understanding is cultivated through training on an extensive corpus of natural language data, encompassing a distinctive dataset composed of 400 million training images paired with their text descriptions, sourced abundantly from the internet.

In essence, CLIP is an enhanced image classification model characterized by heightened accuracy and efficiency, heralding a transformative era in the field of multimodal learning.

1) ZERO-SHOT LEARNING USING CLIP

In the proposed solution for meme detection and classification as hateful or non-hateful, the concept of ''zero-shot'' image classification has been leveraged. This approach is particularly advantageous as it allows us to generalize and make predictions on unseen labels without the necessity of specific training for each class. Traditional machine learning models are typically confined to learning and excelling at a single pre-defined task. For example, an image classifier trained exclusively on categorizing dogs and cats may perform well within that specific scope. However, models like CLIP distinguish themselves by possessing the capability to excel at tasks for which they haven't undergone explicit training. This phenomenon is encapsulated by the term ''zero-shot learning.'' Here, the model employs generalization to predict a class that has not been encountered in the training data.

This makes it an ideal candidate for the proposed solution, since the specific nature of hateful and non-hateful memes may vary widely and evolve over time. This adaptability contributes to the robustness of our meme detection framework.

2) APPROACH

Scaling an elementary pre-training task is necessary to attain competitive zero-shot performance on a wide range of image classification datasets. To achieve this, the CLIP model must be trained to recognize a wide range of visual concepts in images and connect them to their names. This is done by applying contrastive learning to a large dataset of image-text pairs. This information is utilized to determine which of 32,768 randomly chosen text descriptions a given image was accurately paired with in our dataset.

a: CONTRASTIVE REPRESENTATION LEARNING

The novelty of the CLIP model lies in its utilization of a contrastive training strategy, a paradigm that leverages positive (image-text pairs) and negative (other images and text) samples to train a scoring function, thereby generating meaningful representations of the data. Through this innovative technique, CLIP is trained to understand that similar representations should converge in the latent space, while dissimilar ones should exhibit considerable separation [17]. Within the model's architecture, encompassing both image and text encoders, the process of contrastive training involves the labeling of image-text pairs, followed by their embedding with various ''objects'' to learn abstracts in the data. This

facilitates the training of a zero-shot classifier on the resulting image and text embeddings.

Deep learning has extensively used contrastive pre-training. One explanation for this is that contrastive pre-training followed by supervised fine-tuning is a paradigm that is more label-efficient and enhances the effectiveness of labeled data. During pre-training, unlabeled images are effectively clustered together in the latent space, resulting in precise decision boundaries between distinct classes. Subsequent supervised fine-tuning, based on this clustering, consistently outperforms random initialization. It is also a better approach because it not only captures shared information from multiple sources, such as images and text, but also maximizes mutual information [18]. Therefore, the adoption of a model rooted in the contrastive approach aligns seamlessly with our research objectives.

b: COSINE SIMILARITY

In an ideal world, the vector representations of text and its corresponding image should be equal. The similarity between the embedded representations for each text and each image represents the ''goodness'' of our model. Similarly, the ''Badness'' is measured by the dissimilarity between them. An optimal model has maximized goodness and minimized badness.

To assess this ''goodness'' we require a method of computing the distance between the image and text vectors. The Euclidean distance, often known as the straight-line distance, is a wonderful option for determining the separation between two points in 2 or 3 dimensions. All points, however, tend to be far apart by the Euclidean measure in a large dimensional space. Hence, the angle between vectors is a more useful metric in higher dimensions. Thus we use the cosine similarity which calculates the cosine of the angle between two vectors [19], to determine the similarity between them in our model as can be seen in Figure 1.

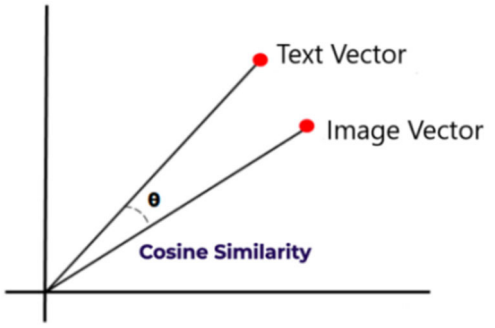

FIGURE 1. Cosine similarity.

A greater value of the cosine distance is produced by more comparable vectors. The dot product of the vectors is used for computation. When not using unit vectors, we must either normalize the vectors or divide the product to the normed vectors. Equation (1) gives the cosine similarity value ($cos\theta$) between given input vectors a and b shown in Eqn. (1).

$$cos\,(\theta) = \frac{(a.b)}{||a||\,||b||} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2}\sqrt{\sum_1^n b_i^2}} \tag{1}$$

c: SOFTMAX FUNCTION

When performing supervised categorization in the contrastive training of the CLIP model utilized, the InfoNCE Loss optimization function is often applied after a softmax function has been applied to the network outputs. Using a vector of real values, the softmax function restricts their range to lie within 0 and 1, with the total of all the numbers equaling 1. Another characteristic of softmax is that it ensures that any one of the values is often much larger than the others, consequently we get a positive example (closest vector) that is significantly larger than the random ones and can be easily identified [20]. Therefore, we first take the softmax of the values and then the negative log of the labeled category to calculate the loss for categorical cross-entropy. Refer to ''(2)'' for the Softmax Function equation.

$$\sigma\,(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} for i = 1, \ldots, K, \quad and$$
$$z = (z_1, \ldots, z_k) \in R^k \tag{2}$$

where,

$$\sigma_i = Softmax\ Function$$
$$z = Input\ Vector$$
$$e^{z_i} = Standard\ exponential\ function\ for\ input\ vector$$
$$K = Number\ of\ classes\ in\ the\ multi - class\ classifier$$
$$e^{z_j} = Standard\ exponential\ function\ for\ output\ vector$$

d: InfoNCE CONTRASTIVE LOSS FUNCTION

The contrastive loss function compares the distance between a sample and the network's output for a positive example of the same class to its distance from a negative example. If positive samples are encoded to similar (closer) representations and negative samples to dissimilar (farther) representations, the loss is low. This is achieved by taking the cosine distances between the vectors and treating the resulting distances as the prediction probabilities of a standard categorization network. The popular loss function we use for contrastive learning in this paper is InfoNCE (NCE is an acronym for Noise-Contrastive Estimation) which is an altered variant of the cross-entropy loss function [21]. The similarity of positive pairs is maximized while that of negative pairs is minimized using this function. In ''(3)'', as shown at the bottom of the next page, z^a, z^p, and z^n represent the anchor, positive, and negative embeddings. According to self-supervised learning, we have one positive sample and many negatives (N). Through the cos_sim function, the vector cosine similarity is assessed. We aim to maximize the cosine similarity between z^a and z^p, bringing them closer together by using this function. The reverse is true for z^a

and z^n. Additionally, there is a temperature hyperparameter, designated by τ, which regulates the degree of penalty for harder negative samples. Harder negatives incur an increased penalty at lower temperatures.

The numerator here is essentially the output of a positive pair, and the denominator is the sum of all values of positive and negative pairs. Ultimately, this simple loss forces the positive pairs to have a greater value, closer to 1 (as pushing the log term to 1 will make $-\log(1) = 0$, which is the optimal loss) and the negative pairs further apart (closer to 0). This loss function can also be interpreted geometrically. Since z^a, z^p, and z^n are high-dimensional latent vectors and normalized, they can be simply seen as points on a hyper-sphere. The cosine similarity between any two of these is then just the Euclidean distance between them shows in Eqn. (4).

$$similarity = cos\,(\theta) = \frac{(A.B)}{||A||\,||B||} \tag{4}$$

As seen in "(4)", for two normalized vectors A and B, since cosine function is inversely proportional to the angle, the bigger their similarity, the smaller the angle is or nearer they are. Thus, as shown in Figure 2, if the two blue vectors form a positive pair, they would be getting nearer like the red one through learning. Otherwise, if they form a negative pair, they would be split apart like black ones through learning.

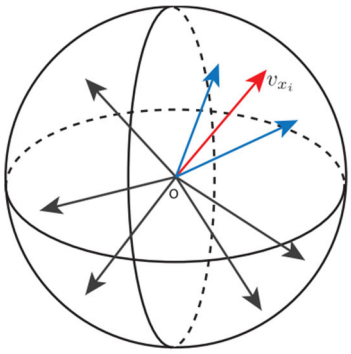

FIGURE 2. Similarity between vector.

e: InfoNCE NETWORK ARCHITECTURE

The InfoNCE contrastive learning method has a network architecture as illustrated in Figure 3. The input x is augmented to x_i and x_j with different augmentation operators. Then they are passed forward through the neural network f to get representations h_i and h_j, on which nonlinear transformation g is performed to get z_i (text vector) and z_j (image vector). Finally, contrastive loss is evaluated on z_i and z_j to optimize f and g. During training, x_i and x_j, if from the same input x, are used as a positive pair, and if from a different input x, they are used as a negative pair.

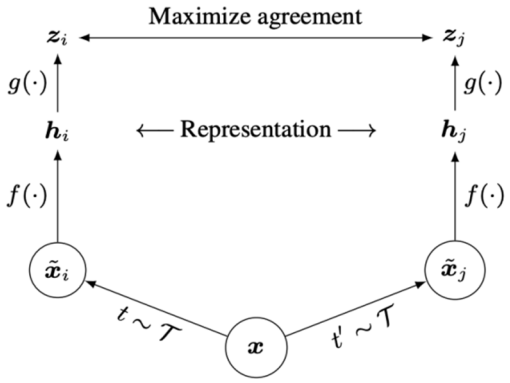

FIGURE 3. InfoNCE network architecture.

3) CLIP MODEL ARCHITECTURE

Multi-Modal architectures leverage more than one domain to learn a specific task. CLIP is a novel architecture that integrates computer vision and natural language processing. Its architecture is designed in a way that both the visual image and the text caption in a multimodal meme can be analyzed simultaneously to extract the text and image embeddings [22], [23], [24]. A text encoder and an image encoder are the two primary parts of its architecture. To predict the most suitable pairings in a batch of training (image, text) examples, these two encoders are trained jointly-

- The core of the text encoder is a transformer model, its base size requires 63 million parameters, 12 layers, and a 512-wide model with 8 attention heads in order to obtain the text features.
- The image encoder, on the other hand, uses both a Vision Transformer (ViT) and a ResNet50 as its backbone, responsible for generating the feature representation of the image.

4) WORKING OF THE CLIP MODEL

The image and text pairs need to be embedded for them to be linked to one another. If we had one cat and two dogs, for instance, we might represent that information as a dot on a graph, embedding the data on the X-Y grid (Euclidean space), as depicted in Figure 4. Both the text and the images work in a similar manner.

Of the two sub-models composing CLIP, the image encoder embeds images into a mathematical space while the text encoder embeds words into one. Then, using contrastive pre-training, CLIP is trained to predict how probable it is that the image corresponds to the text. When compared to other approaches, CLIP is four times more effective at this zero-shot image classification.

$$L_{InfoNCE} = -log\frac{\exp\left(\cos\_sim(z^a, z^p)/\tau\right)}{\exp\left(\cos\_sim(z^a, z^p)/\tau\right) + \sum_{n\in N}\exp\left(\cos\_sim(z^a, z^p)/\tau\right.} \tag{3}$$

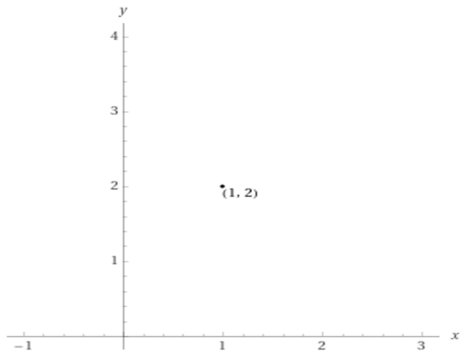

FIGURE 4. Embedding of "1 cat, 2 dogs."

The working of the model consists of three steps: 1) contrastive pre-training, 2) dataset classifier creation from labeled text, 3) application of zero-shot classification.

a: CONTRASTIVE PRE-TRAINING

During this phase, a batch of N (32,768) images paired with their respective descriptions e.g., <image1, text1>, <image2, text2>, <imageN, textN> are processed through the Image and text Encoders simultaneously to obtain their vector representations (embeddings).

A series of purple text cards are being delivered into the text encoder in the example image. Each card's output would be a list of numbers. "Pepper the Aussie dog", for instance, would enter the text encoder and emerge as a series of digits like (0, 0.2, 0.8). The similar thing takes place with the images: each image will enter the image encoder and come out as a series of integers. The image of Pepper the Australian dog will appear as (0.05, 0.25, 0.7).

The CLIP model is then trained to predict which image embedding belongs to which text embedding in a batch. To achieve this contrastive pre-training method seeks to compute the cosine similarity between every pair of image embeddings (I1, I2…IN) and text embeddings (T1, T2…TN). Over the calculated similarity scores, an optimization is executed by applying a symmetric cross-entropy loss to maximize the cosine similarity between the embeddings of real pairs in the batch while minimizing the cosine similarity between the embeddings of incorrect pairings.

The step-by-step procedure is -

- N pairs of "image-text" in batch are sent into the model.
- The Image Encoder computes an image vector for each image in the batch. The I1 vector is represented by the first image, I2 by the second, and so on. The size of each vector is N and N is the latent dimension's size. As a result, N*N matrix is the outcome of this stage.
- Similarly, the text descriptions are transformed into text embeddings (T1, T2 …TN), producing a N*N matrix.
- Finally, we multiply those matrices and calculate the cosine similarities for every single pair of image and text description. This produces an N*N matrix as shown.
- The objective is to achieve the highest possible cosine similarity along the diagonal, which corresponds to the correct image-text embedding pairs (the actual image-text pairs that are maximally near and where i=j) - I1,T1> and I2,T2>.

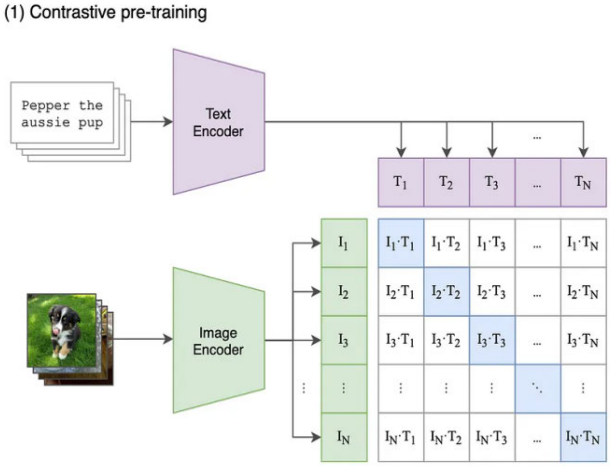

FIGURE 5. Contrastive pre-training phase.

- The light blue squares in Figure 5 stand in for these pairs where the text and image coincide. As an illustration, T1 and I1 are the embedded forms of the first text and first picture, respectively. The highest cosine similarity between I1 and T1 is what we're aiming for. The same thing is desired for I2, T2, and all other light blue squares. The greater these cosine similarities, the more "goodness" our model possesses.
- The cosine similarities of off-diagonal (where i≠j) elements that are dissimilar pairs <I1, T2>, <I1, T3> …<Ii, Tj> are minimized in a contrastive manner, separating the actual image from all the other incorrect text descriptions (for e.g I1 image is described by T1 and not by T2, T2,T3 etc).
- The grey squares in Figure 5 show where the text and image are out of alignment. For instance, T1 might be the text "pepper the aussie pup" while I2 might be a picture of a raccoon. Since "Pepper the Aussie pup" measures "badness," the cosine similarity between this image (I2) and the words "Pepper the Aussie pup" should be quite low.
- The model then uses the symmetric cross-entropy loss as its optimization objective which corresponds to the InfoNCE loss. This type of loss minimizes both the image-to-text direction as well as the text-to-image direction as the contrastive loss matrix keeps both the <I1,T2> and <I2,T1> cosine similarities.

b: CREATE DATASET CLASSIFIER FROM LABEL TEXT

This step encodes all the labels/objects in the following context format: "a photo of a {object}. The vector representation of each context is generated from the text encoder.

It is difficult to find robust datasets with paired image-textual descriptions. Most public datasets, such as CIFAR, are images with just one-single-word labels — these labels are the target class. But CLIP was created to use full textual descriptions. To overcome this discrepancy, using some feature engineering: Single word labels, such as a bird, or a car are converted to sentences. If we have dog, car, and plane as the classes of the dataset, we will output the following context representations:

- a photo of a dog
- a photo of a car
- a photo of a plane

c: USE OF ZERO-SHOT PREDICTION

To perform zero-shot class classification, the image is sent to the encoder, which then conducts a similarity search to determine which text matches the image from the entire batch. For example, the text encoder will contain a batch of "a photo of a dog," "a photo of a car," etc., and CLIP uses these names of all the classes in the dataset (output of section II) as text pairings to predict which image vector corresponds to which text vector or the most probable (text, image) pair. The most similar text prompt is selected as the prediction after computing the pair wise cosine similarities between the image and the text embeddings as can be seen in Figure 6.

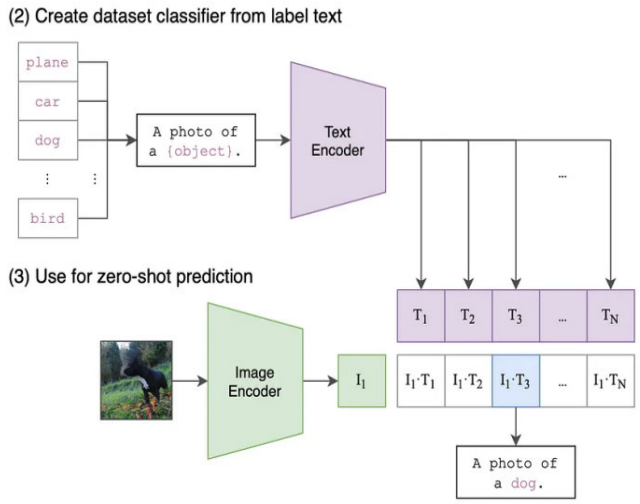

FIGURE 6. Zero-shot classification.

The CLIP model may be demonstrated, as in Figure 7- it is accurately predicting the dog by maximizing the similarity between the word dog and the visual data.

B. TECHNICAL PROCEDURE

The proposed solution utilizes the CLIP multimodal model in Python language using PyTorch machine learning framework and Torchvision library to better understand and classify the multimodal hateful memes involving images and text. It makes use of a pre-trained CLIP model to create a custom classifier without any training required. The generated hateful memes detector achieves competitive results with supervised models baseline using zero-shot classification.

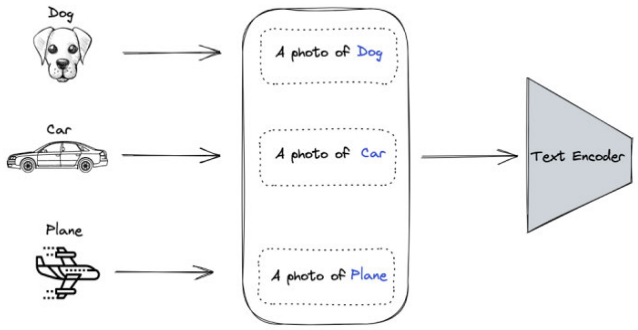

FIGURE 7. Image classification using CLIP.

1) DATASET

The Dataset chosen to implement the Multimodal Hateful Memes Detection is the Facebook Hateful Meme Dataset (https://ai.meta.com/blog/hateful-memes-challenge-and-data-set/). This dataset was produced by Facebook AI with the express purpose of assisting in the creation of new methods to detect multimodal hate speech. It is challenging for machines to comprehend this content since it integrates multiple modalities - text and images. The dataset includes 10,000+ novel, multimodal examples of memes, each of which includes an image and an OCR sentence in it (refer Figure 9). Memes can be classified into two categories for the purposes of this challenge: non-hateful and hateful. Fully balanced, the validation and test sets each contain 5% and 10% of the data (Table 1). The remaining data, which consists of 36% hateful memes and 64% non-hateful memes, is used as a train set. The images in the dataset are all licensed from Getty Images and span a wide range of both attacks (such as encouraging violence or depicting groups as criminals) and protected categories (such as religion, gender, and sexual orientation). The memes in this dataset were chosen in a way that only multimodal models can successfully classify them, making it difficult for strictly unimodal classifiers to do so. When taken separately, the text phrase and the image in each meme are harmless but when taken into consideration, the meme's semantic content becomes offensive.

TABLE 1. Facebook hateful Memes dataset splits.

Input Type	Splits
Validation Set	5%
Test Set	10%
Multimodal Hateful Input	30%
Multimodal Non-Hateful Input	54%
Benign Confounders	10%

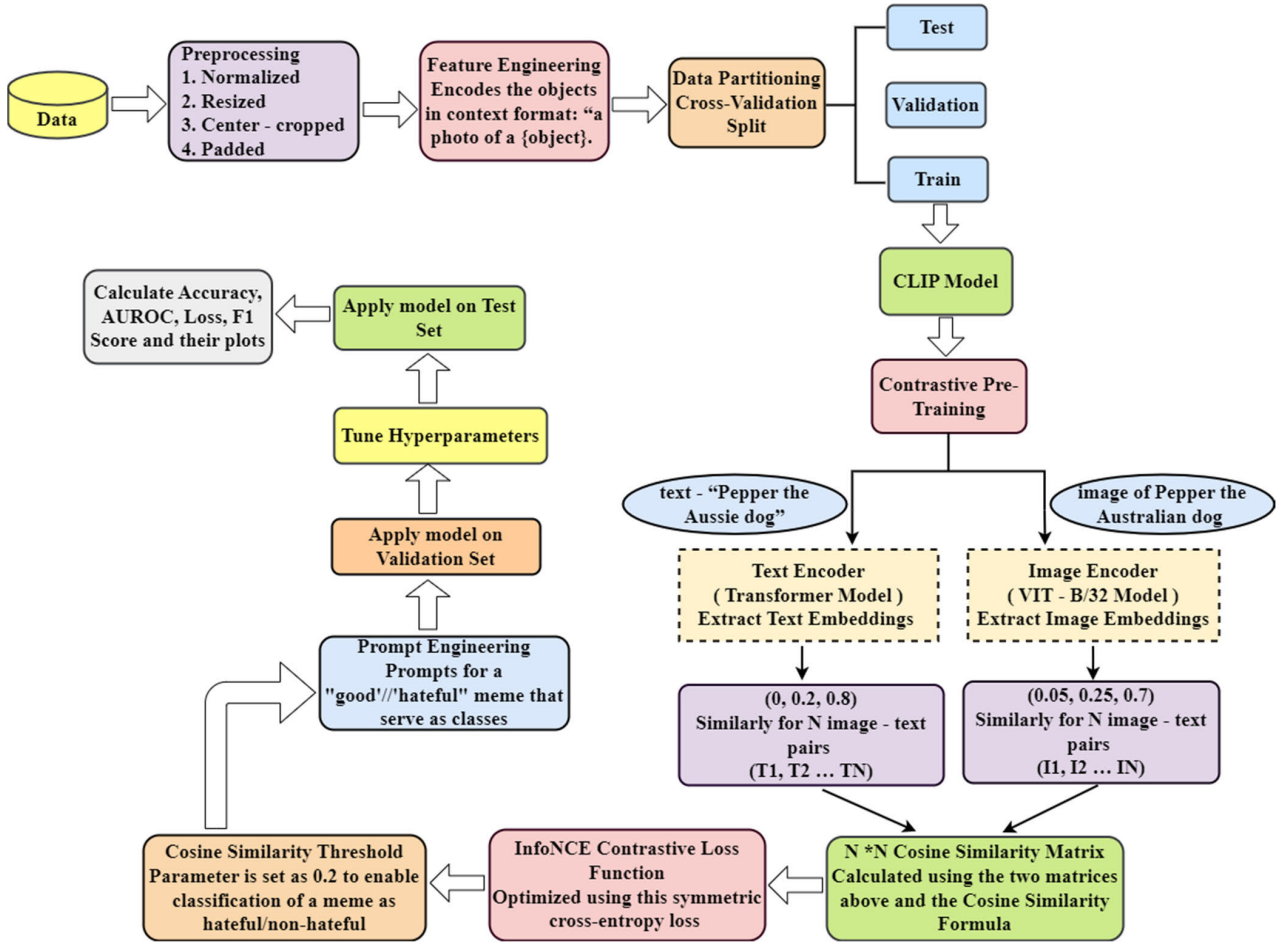

FIGURE 8. Flowchart of the technical procedure.

FIGURE 9. Facebook hateful Meme dataset samples.

FIGURE 10. Benign confounders in the dataset.

The dataset is specifically made to address issues that are frequently encountered in AI research, namely the lack of examples that would enable machines to learn to avoid false positives. This is accomplished by introducing memes in the dataset that resemble offensive examples but are innocuous. These challenging cases, referred to as benign confounders (Figure 10), are included to address potential biases in classification systems and the development of systems that avoid false positives.

2) LOADING THE CLIP MODEL

We load the model and the torchvision transformation pipeline that are needed by it after installing and importing the CLIP model, its weights, tokenizer image processor, and related libraries from OpenAI. The image encoder is either a Vision Transformer (ViT) or a ResNet version like ResNet50, whereas the text encoder is a Transformer. For the goal of identifying the hateful memes, we use the ViT-B/32 as the image encoder. The command **clip.available_models** as shown in Figure 11 can be used to view the image encoders that are offered.

3) EXTRACTING IMAGE EMBEDDINGS

Each image goes through preprocessing before being fed into the Image Encoder. First, the dataset's mean and standard

```
import clip
clip.available_models()
```

```
['RN50',
 'RN101',
 'RN50x4',
 'RN50x16',
 'RN50x64',
 'ViT-B/32',
 'ViT-B/16',
 'ViT-L/14',
 'ViT-L/14@336px']
```

```
device = "cuda" if torch.cuda.is_available() else "cpu"
model, preprocess = clip.load("ViT-B/32", device)
model.cuda().eval()
input_resolution = model.visual.input_resolution
context_length = model.context_length
vocab_size = model.vocab_size
```

FIGURE 11. Loading the clip model.

```
preprocess
```

```
Compose(
    Resize(size=224, interpolation=bicubic, max_size=None, antialias=None)
    CenterCrop(size=(224, 224))
    <function _convert_image_to_rgb at 0x7f5560031950>
    ToTensor()
    Normalize(mean=(0.48145466, 0.4578275, 0.40821073), std=(0.26862954, 0.261
```

FIGURE 12. Extracting the image features.

deviation are used to normalize the input images' pixel intensity. They are then center cropped, normalized, and resized to comply with the image resolution that the image encoder requires. The image is preprocessed (refer Figure 12 for pseudo code) and then sent to the Image Encoder, which produces a 1×512 image embedding tensor as its output [8], [25]. This preprocessing is executed by a torchvision transform function.

4) EXTRACTING TEXT EMBEDDINGS

First, a case-insensitive text tokenizer that is invoked with clip.tokenize() processes the text labels, converting the label words into numeric values. To meet the requirements of the Text Encoder, the outputs are by default padded to a length of 77 tokens. As a result, a padded tensor of size $N \times 77$ (Figure 13) is created (N is the number of classes, which equals 2×77 in binary classification), and this is used as input for the Text Encoder. Following that, the Text Encoder converts the tensor into a $N \times 512$ tensor of text embeddings, where each class is represented by a single vector. The **zmodel.encode_text**() method can be used to encode text and retrieve embedding.

5) CALCULATING AND PLOTTING THE COSINE SIMILARITY MATRIX

We must first identify the relationship between the text feature and the image features before we can label a meme

```
text_tokens = clip.tokenize([desc for desc in texts]).to(device)
with torch.no_grad():
    image_features = model.encode_image(image_input).float()
    text_features = model.encode_text(text_tokens).float()
```

```
tensor([[49406,  3306,  1002,   256, 49407,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0]])
```

FIGURE 13. Extracting the text features & padded tensor.

as hateful. With a potent multimodal model like CLIP, we employ the cosine similarity distance metric to compute the degree of similarity between the various modalities i.e., each text and image encoding.

The model is fed with 8 example images and their associated texts, compute the dot products for each pair, and compare the similarity between the corresponding features. The model () calculates the cosine similarity (refer Figure 14) between the corresponding image and text features and multiplies them by 100 to generate **image_logits,** by passing the preprocessed image and text inputs through the image and text encoders. The logits are then normalized into a list of probability distributions for each class. The class with the highest probability (thus highest similarity score) is then set as the predicted class.

```
image_features /= image_features.norm(dim=-1, keepdim=True)
text_features /= text_features.norm(dim=-1, keepdim=True)
similarity = text_features.cpu().numpy() @ image_features.cpu().numpy()
```

FIGURE 14. Cosine similarity matrix pseudocode.

6) CLASSIFYING THE MEME AS HATEFUL OR NON-HATEFUL

We set the threshold for cosine similarity between the image and its text to be 0.2. This means that, if the cosine similarity between the image and its text falls below 0.2, we directly classify it as non-hateful. This is because, even if the model pairs the image to the hateful description, according to the values obtained in the cosine similarity matrix, we can see that for similarity scores below 0.2, the text features does not relate in context to the hateful features of the image and thus combined cannot be hateful.

If the cosine similarity between the image and its text is greater than 0.2, then we classify it based on whether the model associates the image to the good meme or hateful meme description as shown in Figure 15. This approach allows us to combine both the contributions of text and

```
if similarity < 0.2:
    predicted_label = 0
    predicted.append(predicted_label)
else:
    hate_similarity = (image_features.cpu() @ F_text_features
    if hate_similarity[0][0] > hate_similarity[0][1]:
        predicted_label = 0
        predicted.append(predicted_label)
    else:
        predicted_label = 1
        predicted.append(predicted_label)
```

FIGURE 15. Multimodal Meme classification.

image feaures towards detecting hatefulness of the input meme.

7) PROMPT ENGINEERING

Without any training or fine-tuning, a powerful model like CLIP can produce zero-shot predictions. To achieve that, we offer the model some text prompts [26], [27]. These text labels or prompts are encoded by the CLIP classifier into a learned latent space, and their similarity to the image latent space is assessed. The classifier's performance may be altered by changing the language of the prompts because different text embeddings can have a different effect. We produce written descriptions for a ''good meme'' and a ''hateful meme'' as can be seen in Figure 16, that serve as classes for our dataset to classify an input image meme as hateful or non-hateful.

```
#Create text labels that act as classes for our dataset
descriptions = {
      'good meme': 'a nonhateful meme that is good',
      'hateful meme': 'a hateful meme containing racism, sexism, nationality,
lity'
}

text_labels = [descriptions['good meme'],descriptions['hateful meme']]
text_tokens = clip.tokenize([desc for desc in text_labels]).to(device)
```

FIGURE 16. Text prompts.

8) CALCULATING THE ACCURACY OF THE MODEL

To evaluate the model's performance, we compare its performance to a validation dataset—a set of data on which it has not been trained. The most popular evaluation metric, ''accuracy,'' is calculated as the proportion of correctly classified images to all the images in your data set. We find the accuracy of this model to be 57.8% which is quite high for a model that is pre- trained and predicts results using generalization of unseen labels in zero shot classification. We can improve the accuracy of the model through prompt engineering by changing the text descriptions for labels being fed into the model.

9) TESTING THE MODEL AND MAKING PREDICTIONS

After successfully implementing the CLIP model, it is ready to predict outcomes for unseen data sets. Hence, we feed the test data which consists of only images and texts and no labels into it. The model returns the expected output of all the input images being successfully classified into their predicted labels according to their probability scores.

IV. RESULTS AND OUTPUT

In this paper we used the Facebook Hateful Meme dataset (https://ai.meta.com/blog/hateful-memes-challenge-and-data-set/) to detect hate speech in the multimodal image text combinations of memes by implementing the CLIP model. On feeding random unseen memes as input to the model, we can see that the CLIP model gave us highly accurate results, by predicting and classifying each combination of text and image that falls into the category of hate speech as a ''Hateful'' meme correctly. Also, no non-hateful meme out of all the inputs is falsely classified as hateful i.e., there are no false positive outputs in the result. Therefore, it is seen the model is accurate and is working successfully by predicting correct outcomes.

A. PREDICTED OUTCOMES AND ANALYSIS

The cosine similarity matrix obtained while applying the CLIP model to the chosen dataset can be observed in Figure 16. This similarity matrix illustrates a visual representation of the relationship between the texts and images in the hateful memes dataset by calculating the cosine similarity between the image and text features. As seen in the figure, the highest possible cosine similarity between the image and text pairs is achieved along the diagonal, or yellow squares, since they have the highest dot product values. In other words, these image text pairs are the closest to each other in what they are describing or have the maximum correlation. On the other hand, the blue and purple squares signify that the corresponding image text pair is completely out of alignment or that the given statement is completely different from what is being shown in the picture.

Then the accurately predicted outcomes of the validation set can be seen below, as the memes containing hate speech are correctly identified and classified as ''Hateful Memes'' in Figures 18–23. There are a total of 358 memes classified accurately as hateful out of the 1000 unknown input images in the test dataset. The first 20 of these ''happy memes'' are shown in a grid in Figure 17. These precise classification outcomes are a result of setting the threshold for cosine similarity value accurately and fine-tuning the CLIP model using prompt engineering. In the proposed solution, the threshold of cosine similarity between the image and its text is set at 0.2 for precise predictions. As a result, all image text pairs with a cosine similarity value below 0.2 are automatically classified as non-hateful. For values above

FIGURE 17. Cosine similarity matrix.

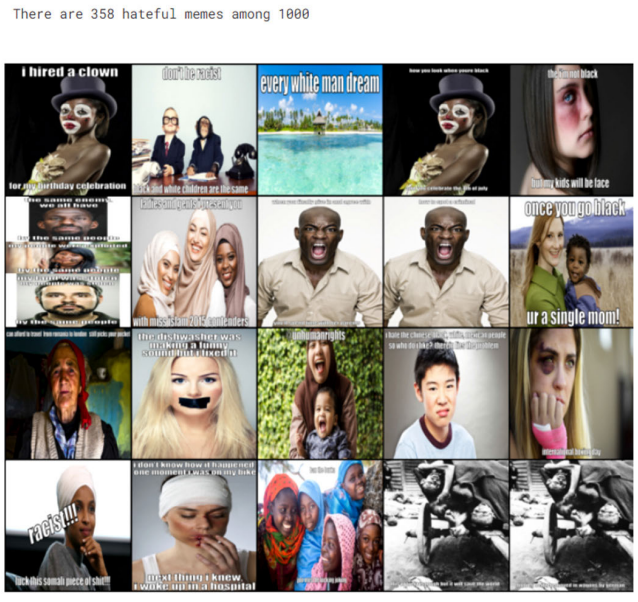

FIGURE 18. Accurately predicted outcomes.

FIGURE 19. "Hateful Meme."

0.2, the model performs the classification process according to the given text prompts for a good and a hateful meme. In this case, the prompts provided are - Good Meme: A non-hateful meme that is good." and Hateful Meme: "A hateful meme containing racism, sexism, nationality, religion, and disability. As seen from the results, the CLIP model classified the unknown dataset images with precision based on the key words of the given prompts.

- Some of the memes classified as hateful are –

FIGURE 20. "Hateful Meme."

FIGURE 21. "Hateful Meme."

FIGURE 22. "Hateful Meme."

B. MODEL EVALUATION PARAMETERS

The evaluation metrics, including accuracy, AUROC, Loss%, and F1-score were computed for the proposed CLIP model, and the results are tabulated in Table 2. The model's Accuracy and Loss graphs have also been plotted (refer

FIGURE 23. "Hateful Meme."

TABLE 2. Evaluation parameters of the CLIP model.

Parameter	Value%
Accuracy	87.42
Loss %	35.70
AUROC	88.35
F1 Score	90.13

Figures 24 to 27). The Model Accuracy Plot with the validation accuracy line being an increasing curve shows that the model performs accurately with a score of 87.42%. The Model's Loss Plot with the graph dipping shows that the loss calculated during its performance is low with a base line of 0.357. Since the model loss is low, the model accuracy is high as they are inversely proportional to each other, leading to the proposed model being an optimal and precise solution for the classification of memes as hate speech. The high AUROC value (or the model's ability to differentiate between positive and false-positive samples) and f1 score also indicate the high efficiency of the model implemented.

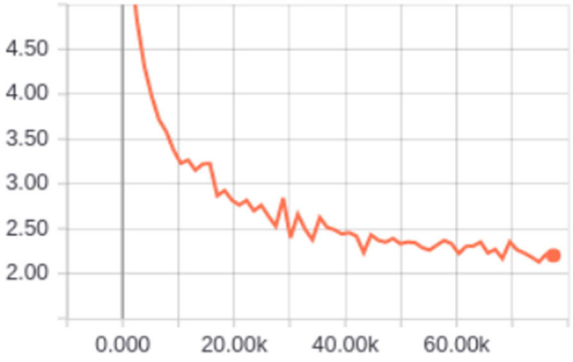

FIGURE 24. Training cross entropy loss.

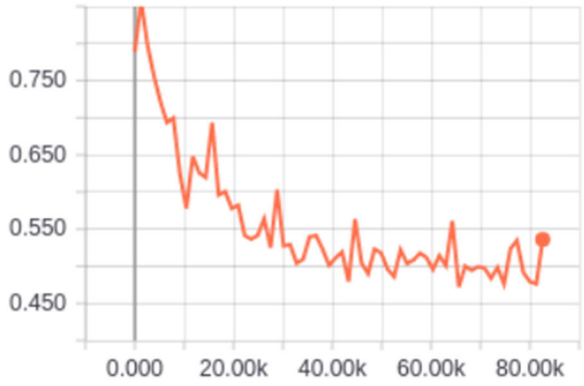

FIGURE 25. Validation cross entropy loss.

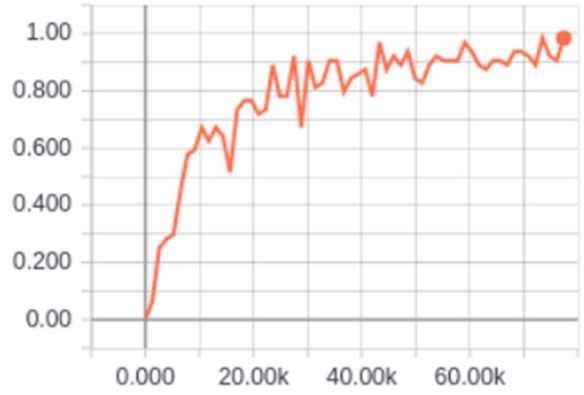

FIGURE 26. Training accuracy.

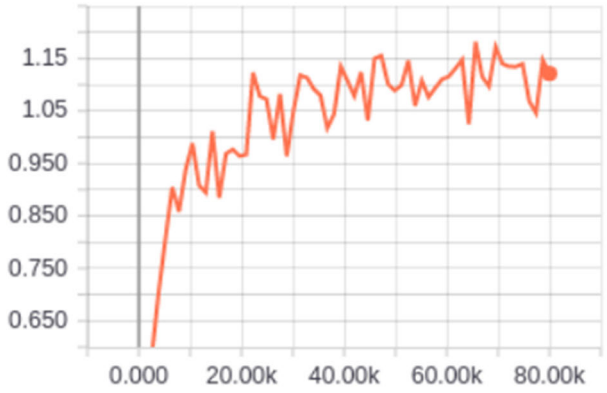

FIGURE 27. Validation accuracy.

C. COMPARATIVE ANALYSIS

The classification of hateful memes can be quite a challenging task due to the dual nature of the data that needs to be extracted from the input images. For the successful and accurate prediction of hateful memes, both the image and text features need to be extracted from the input meme, which will allow us to combine both the contributions of the text and image embeddings towards detecting the hatefulness of the input. However, far fewer studies focus on the multimodal representation of data, namely the information that consists of multiple channels, since most classification tasks are

"unimodal" or can only extract and learn in one mode, either texts or images. Hence, the conventional methods of classification of hateful memes rely on unimodal models, which prove to be very less effective as training a model using only one dimension out of images, text, or video is an extremely difficult endeavor. Thus, this task requires a multimodal model that includes text and visuals that are trained successfully and simultaneously for accurate results. There have been a few previous works that have taken this direction and implemented multimodal study of hateful memes using methods such as visual question answering, vision language pre-training models, and encoders based on convolutional neural networks, apart from making use of unimodal techniques.

Zhong et al. proposes a new model that combines multimodal features with rules, achieving the highest accuracy of 86.8% [28]. It leverages a specific dataset developed by Facebook with over 10,000 memes. The specimens encompass memes in the percentages of 10% unimodal hate, 20% benign image confounder, 20% benign text confounder, and 10% random non-hateful. Here, a clustering technique based on perceptual hash is used to group the meme images together. By using a straightforward comparison on their strings, the memes are grouped into groups, including "3-tuple," "2-tuple," "unimodal hate," etc. The "3-tuple," for example, is made up of 3 memes, with the first meme having an image like the second meme and text equivalent to the third meme. The second meme and the third meme, however, are not connected. The labels for a "3-tuple" consist of 1, 0, and 0, where 1 denotes hate and 0 denotes non-hatred, while the labels for a "2-tuple" consist of 1 and 0. From the analysis above, there were rules formulated, such as Rule 1, where the hatred probability for samples in a "3-tuple" was set to (1, 0, 0). and Rule 2, where in the case of samples in "2-tuple," the hateful probabilities were set to (1,0), with the larger hateful probability being adjusted to 1.

Alternatively, Ahmed et al. explores the use of unimodal text and image models, such as Bert, LSTM, VGG16, Resnet50, SE-Resnet50, and XSE-Resnet architectures, and combines them into multimodal models for predicting hateful memes with evaluation metrics such as the AUC-ROC score, F1 score, and accuracy score [29]. The dataset selected for the endeavor was also the "Hateful Memes Challenge," released by Facebook AI, but yielded results of a maximum prediction accuracy of 66.3% in classifying memes as hateful or non-hateful.

Fan et al. utilises data from Meta's Hateful Meme Detection Challenge and builds three models, with their best model, VisualBERT with external feature extraction, achieving a 62.4% accuracy [30]. Kiela et al. highlights the difficulty of the hateful meme detection task, with state-of-the-art methods performing poorly compared to humans (64.73% vs. 84.7% accuracy) [9]. It then evaluates a variety of models—unimodal models and multimodal models—that were unimodally pretrained (a BERT model combined with a ResNet) on the hateful memes dataset and concludes that the random and majority-class baselines lie at 50 AUROC for the unimodal text-only or visual-only classifiers. Multimodal models such as VilBERT are also able to achieve a maximum of 72 AUROC.

Lastly, Sethi et al. investigates the classification of hateful memes using pre-trained models like VGG19 and Xception, combined with machine learning models like support vector machines and Naïve Bayes [31]. They achieve the highest f1-score of 0.584 using an integrated stacked model technique. Table 3 shows the results of a comparison between the evaluation metrics (AUROC and accuracy score) achieved by various models utilized in several state-of-the-art approaches for the classification of hateful memes [32], [33], [34], [35].

TABLE 3. Performance comparison of proposed and existing models.

	Model	Accuracy%	AUROC%
Unimodal Models	Image-Grid	50.67	52.33
	Image-Region	52.53	57.24
	Text BERT	58.27	65.05
Multimodal Models	MMBT-Region	64.75	72.62
	ViLBERT	63.16	72.17
	Visual BERT	65.01	74.14
	CLIP MODEL	87.42	88.35

The results of this proposed study demonstrate that the CLIP model, fine-tuned with prompt engineering, can achieve an accuracy rate of 87.42% and an AUROC of 88.35 in the classification of hateful memes when implemented on the Facebook Hateful Memes Dataset. This represents a substantial improvement compared to previous studies that employed machine learning methods for the detection of hate speech in memes.

As shown in Table 3, the proposed CLIP model outperforms all other models in terms of accuracy.

In contrast with previous works, the implementation of the CLIP model also presents us with an easy, feasible option for the classification process since this model is already pre-trained and does not need to go through a highly time-consuming process of training over large sets of data. This not only saves time but also saves the effort of a large data accumulation process. The proposed model also does not require a complex combination of rules for attaining high accuracy and is thus a simpler method of achieving efficiency.

The approach taken in this study is also able to provide better results than previous works since the dataset chosen—

the Facebook Hateful Memes Dataset—is a more difficult dataset containing several false positive examples and benign confounders and is paired with prompt engineering. Incorporation of prompt engineering is a deliberate and systematic curation of accurate prompts to serve as classes for increasing the accuracy in meme classification. The strategic pairing of this approach with the intricacies of the Facebook Hateful Memes Dataset aims to mitigate the impact of false positives and confounding variables, ultimately elevating the discriminatory capabilities of the proposed model and making it challenging to rely on unimodal signals, resulting in the success of multimodal models. This makes the proposed strategy more efficient and precise as compared to others.

V. FUTURE SCOPE

For detecting hateful memes our model that included texts and visuals being trained simultaneously was able to provide successful results with precision. Its accuracy can be further improved to give extremely accurate results, by training the model especially for a particular dataset since the model that is utilized presently is pre-trained contrastively for a general dataset. But training of multimodal data is an extremely cumbersome process and there is a lack of easy access to the relevant hardware and software devices required for it. Social media's development over the past few decades has made a wealth of information readily accessible online which makes a multimodal dataset such as the one used extremely complex and very large. The training process will thus require a lot of GPU hours and memory before the model can be used to successfully classify hate speech. But in the future, this can be made possible with access to better technology and time to develop, test, and improve models used for hate speech detection.

VI. CONCLUSION

This paper aims to develop a novel and efficient architecture for detecting and classifying multimodal hate speech in memes circulating through social media. The suggested strategy for this is making use of OpenAI's latest multimodal model - CLIP, to better understand multimodal hate speech in memes that contain both visual images and text captions. The CLIP model analyses the image and its accompanying text to determine whether the two modalities taken together are hateful or not. The ''Facebook Hateful Meme Dataset,'' which consists of 10,000 examples of new multimodal memes (text + image) created by Facebook AI, is utilized as the dataset for the proposed method. The implemented model has been able to achieve an accuracy of 87.42% in recognizing hateful memes. This scope of this study can be further extended to filter out these memes from the social media platforms to keep a check on the hatred spreading through such content online. This will help control the hate spread against minority communities and diminish any form of discrimination such as racism or sexism through cyber platforms. It will also curb cyber bullying and hate speech on social media generated by trolls using offensive memes. Amidst the incoming network traffic, such hate speech will be recognized and routed so that the user is protected. The perpetrators spreading hate online through such memes can then be identified and punished. Additionally, further research into this suggested method may also aid in improving the accuracy of hate speech detection in memes, by exploring the possibilities of advanced machine learning algorithms in managing multimodal data.

ACKNOWLEDGMENT

The authors acknowledges to the Universiti Kebangsaan Malaysia for supporting this work under GUP 2023-010, and Advanced and Innovative Research Laboratory, India, for the technical support.

REFERENCES

[1] R. Richard and G. Giorgi, ''What is a meme, technically speaking?'' *Inf., Commun. Soc.* pp. 1–19, Feb. 2023.

[2] Z. Mansur, N. Omar, and S. Tiun, ''Twitter hate speech detection: A systematic review of methods, taxonomy analysis, challenges, and opportunities,'' *IEEE Access*, vol. 11, pp. 16226–16249, 2023, doi: 10.1109/ACCESS.2023.3239375.

[3] E. K. Boahen, B. E. Bouya-Moko, F. Qamar, and C. Wang, ''A deep learning approach to online social network account compromisation,'' *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 6, pp. 3204–3216, Dec. 2023, doi: 10.1109/TCSS.2022.3199080.

[4] K. Abbas, M. K. Hasan, A. Abbasi, U. A. Mokhtar, A. Khan, S. N. H. S. Abdullah, S. Dong, S. Islam, D. Alboaneen, and F. R. A. Ahmed, ''Predicting the future popularity of academic publications using deep learning by considering it as temporal citation networks,'' *IEEE Access*, vol. 11, pp. 83052–83068, 2023.

[5] H. Hosseinmardi, S. Arredondo Mattson, R. Ibn Rafiq, R. Han, Q. Lv, and S. Mishra, ''Detection of cyberbullying incidents on the Instagram social network,'' 2015, *arXiv:1503.03909*.

[6] H. Zhong, H. Li, A. C. Squicciarini, S. M. Rajtmajer, C. Griffin, D. J. Miller, and C. Caragea, ''Content-driven detection of cyberbullying on the Instagram social network,'' in *Proc. IJCAI*, vol. 16, 2016, pp. 3952–3958.

[7] Y. Chen and F. Pan, ''Multimodal detection of hateful memes by applying a vision-language pre-training model,'' *Plos One*, vol. 17, no. 9, 2022, Art. no. e0274300.

[8] Y. Zhou, Z. Chen, and H. Yang, ''Multimodal learning for hateful memes detection,'' in *Proc. IEEE Int. Conf. Multimedia Expo. Workshops (ICMEW)*, Jul. 2021, pp. 1–6, doi: 10.1109/ICMEW53276.2021.9455994.

[9] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, and D. Testuggine, ''The hateful memes challenge: Detecting hate speech in multimodal memes,'' 2020, *arXiv:2005.04790*.

[10] T. Deshpande and N. Mani, ''An interpretable approach to hateful meme detection,'' in *Proc. Int. Conf. Multimodal Interact.*, New York, NY, USA, Oct. 2021, pp. 723–727, doi: 10.1145/3462244.3479949.

[11] A. A. Ahmed, M. K. Hasan, M. M. Jaber, S. M. Al-Ghuribi, D. H. Abd, W. Khan, A. T. Sadiq, and A. Hussain, ''Arabic text detection using rough set theory: Designing a novel approach,'' *IEEE Access*, vol. 11, pp. 68428–68438, 2023.

[12] A. K. Thakur, F. Ilievski, H. Sandlin, Z. Sourati, L. Luceri, R. Tommasini, and A. Mermoud, ''Multimodal and explainable internet meme classification,'' Dec. 2022, *arXiv:2212.05612*.

[13] A. Radford, J. Wook Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, ''Learning transferable visual models from natural language supervision,'' 2021, *arXiv:2103.00020*.

[14] Y. Qu, X. He, S. Pierson, M. Backes, Y. Zhang, and S. Zannettou, ''On the evolution of (Hateful) memes by means of multimodal contrastive learning,'' in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Francisco, CA, USA, May 2023, pp. 293–310, doi: 10.1109/sp46215.2023.10179315.

[15] K. Abbas, M. K. Hasan, A. Abbasi, S. Dong, T. M. Ghazal, S. N. H. S. Abdullah, A. Khan, D. Alboaneen, F. R. A. Ahmed, T. E. Ahmed, and S. Islam, ''Co-evolving popularity prediction in temporal bipartite networks: A heuristics based model,'' *IEEE Access*, vol. 11, pp. 37546–37559, 2023.

[16] A. Bhandari, ''Bias in AI: A comprehensive examination of factors and improvement strategies,'' *Int. J. Comput. Sci. Eng.*, vol. 10, no. 6, pp. 9–14, Jun. 2023, doi: 10.14445/23488387/ijcse-v10i6p102.

[17] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.
[18] J. Badour and J. A. Brown, "Hateful memes classification using machine learning," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Orlando, FL, USA, Jun. 2021, pp. 1–8, doi: 10.1109/SSCI50451.2021.9659896.
[19] S. Prabhakaran. *Cosine Similarity—Understanding the Math and How it Works (With Python Codes)*. Accessed: Oct. 20, 2023. [Online]. Available: https://www.machinelearningplus.com/nlp/cosine-similarity/
[20] K. E. Koech. *Softmax Activation Function—How it Actually Works*. Towardsdatascience.com. Accessed: Oct. 5, 2023. [Online]. Available: https://towardsdatascience.com/softmax-activation-function-how-it-actually-works-d292d335bd78
[21] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.
[22] M. S. Hee, R. K.-W. Lee, and W.-H. Chong, "On explaining multimodal hateful meme detection models," in *Proc. ACM Web Conf.*, New York, NY, USA, Apr. 2022, pp. 3651–3655, doi: 10.1145/3485447.3512260.
[23] P. Lippe, N. Holla, S. Chandra, S. Rajamanickam, G. Antoniou, E. Shutova, and H. Yannakoudakis, "A multimodal framework for the detection of hateful memes," 2020, *arXiv:2012.12871*.
[24] M. A. Latiffi and M. R. Yaakub, "Sentiment analysis: An enhancement of ontological-based using hybrid machine learning techniques," *Asian J. Inf. Technol.*, vol. 7, pp. 61–69, Dec. 2018.
[25] A. Mukred, D. Singh, and N. S. Mohd Satar, "Examining the influence of perceived need on the adoption of information system in public hospitals in Yemen," *Asia–Pacific J. Inf. Technol. Multimedia*, vol. 9, no. 2, pp. 35–49, Dec. 2020.
[26] R. Cao, R. Ka-Wei Lee, W.-H. Chong, and J. Jiang, "Prompting for multimodal hateful meme classification," 2023, *arXiv:2302.04156*.
[27] I. Memon, R. A. Shaikh, M. K. Hasan, R. Hassan, A. U. Haq, and K. A. Zainol, "Protect mobile travelers information in sensitive region based on fuzzy logic in IoT technology," *Secur. Commun. Netw.*, vol. 2020, pp. 1–12, Nov. 2020.
[28] X. Zhong, "Classification of multimodal hate speech—The winning solution of hateful memes challenge," 2020, *arXiv:2012.01002*.
[29] Md. R. Ahmed, N. Bhadani, and I. Chakraborty, "Hateful meme prediction model using multimodal deep learning," in *Proc. Int. Conf. Comput., Commun. Green Eng. (CCGE)*, Sep. 2021, pp. 1–5.
[30] A. Fan and Y Wu. *Identifying Hateful Memes With Multimodal Classification*. Cs231n.stanford.edu. Accessed: Sep. 10, 2023. [Online]. Available: http://cs231n.stanford.edu/reports/2022/pdfs/66.pdf
[31] A. Sethi, U. Kuchhal, and R. Katarya, "Study of various techniques for the classification of hateful memes," in *Proc. Int. Conf. Recent Trends Electron., Inf., Commun. Technol. (RTEICT)*, Bangalore, India, Aug. 2021, pp. 675–680, doi: 10.1109/RTEICT52294.2021.9573926.
[32] A. Gao, B. Wang, J. Yin, and Y. Tian, "Hateful memes challenge: An enhanced multimodal framework," 2021, *arXiv:2112.11244*.
[33] Y. Chen and F. Pan, "Multimodal detection of hateful memes by applying a vision-language pre-training model," *PLoS One*, vol. 17, no. 9, 2022, Art. no. e0274300, doi: 10.1371/journal.pone.0274300.
[34] Z. Ma, S. Yao, L. Wu, S. Gao, and Y. Zhang, "Hateful memes detection based on multi-task learning," *Mathematics*, vol. 10, no. 23, p. 4525, Nov. 2022, doi: 10.3390/math10234525.
[35] M. G. Constantin, D.-S. Parvu, C. Stanciu, D. Ionascu, and B. Ionescu, "Hateful meme detection with multimodal deep neural networks," in *Proc. Int. Symp. Signals, Circuits Syst. (ISSCS)*, Iasi, Romania, Jul. 2021, pp. 1–4, doi: 10.1109/ISSCS52333.2021.9497374.

GREESHMA ARYA received the B.Tech. and M.Tech. degrees (Hons.) from Dr. A. P. J. Abdul Kalam University, Lucknow, India, and the Ph.D. degree in electronics and communication engineering from Uttarakhand Technical University, Dehradun, India. She is currently an Associate Professor with the Department of Electronics and Communication Engineering, Indira Gandhi Delhi Technical University for Women (IGDTUW) (State Government University), Delhi, India. She has more than 17.5 years of experience in academics and research. She has published more than 35 research articles in various international journals (including SCI, ESCI, Scopus, and ISI indexed). Her areas of interests include wireless sensor networks, wireless communication, renewable energy sources, network security, the Internet of Things, 5G network technology, artificial intelligence, and deep learning.

MOHAMMAD KAMRUL HASAN (Senior Member, IEEE) received the Ph.D. degree in electrical and communication engineering from the Faculty of Engineering, International Islamic University, Malaysia, in 2016. He is currently an Associate Professor and the Head of the Network and Communication Technology Laboratory, Faculty of Information Science and Technology, Center for Cyber Security, Universiti Kebangsaan Malaysia (UKM). He is specialized in elements pertaining to cutting-edge information centric networks, computer networks, data communication and security, mobile network and privacy protection, cyber-physical systems, industrial IoT, transparent AI, and electric vehicles networks. He has published more than 230 indexed papers in ranked journals and conference proceedings. He is a member of the Institution of Engineering and Technology and the Internet Society. He is a Certified Professional Technologist in Malaysia. He has actively participated in many events/workshops/trainings for the IEEE and IEEE Humanity Programs in Malaysia. He served as the Chair for IEEE Student Branch, from 2014 to 2016. He is the general chair, the co-chair, and a speaker of conferences and workshops for the shake of society and academy knowledge building and sharing and learning. He has been contributing and working as a volunteer for underprivileged people for the welfare of society. He is an editorial member in many prestigious high-impact journals, such as IEEE, IET, Elsevier, Frontier, and MDPI.

ASHISH BAGWARI (Senior Member, IEEE) received the B.Tech. (Hons.), M.Tech. (Hons.), and Ph.D. degrees in electronics and communication engineering. He is currently the Head of the Department of Electronics and Communication Engineering, Women Institute of Technology (WIT) (Institute of State Government), Affiliating Institution of Uttarakhand Technical University, Dehradun, India. He has more than 14.5 years of experience in industry, academics, and research. He has published more than 170 research articles in various international journals that also include IEEE international conferences. His areas of interest are cognitive radio networks, mobile communication, sensor networks, wireless, and 5G Communication, digital communication, and mobile ad-hoc networks. He is an active member of various professional societies, such as IEEE, USA. He is also a Senior Member of the Institute of Electronics and Telecommunication Engineers (IETE), India; a Lifetime Member and a Professional Member of the Association for Computing Machinery (ACM); and a member of the Machine Intelligence Research Laboratory Society. He received the Gold Medalist during the master's study. He also received the Best WIT Faculty Award, in 2013 and 2015; the Best Project Guide Award, in 2015; and the Corps of Electrical and Mechanical Engineers Prize from the Institution of Engineers, India (IEI), in December 2015, for his research work. Also, he received the Outstanding Scientist Award 2021 from VDGOOD Technology, Chennai, India, in November 2021; the Dr. A. P. J. Abdul Kalam Life Time Achievement National Award 2022 from National Institute for Socio Economic Development (NISED), Bangalore, India, in June 2022; and the Best Teacher Award-2023 from Veer Madho Singh Bhandari Uttarakhand Technical University (State Government Technical University), Dehradun, in September 2023. He was named in Who's Who in the World 2016 (33rd Edition) and 2017 (34th Edition).

NURHIZAM SAFIE (Member, IEEE) received the master's degree in information technology from UKM, in 1999, the M.B.A. degree from Anglia Ruskin University, U.K., in 2019, and the Ph.D. degree in management information systems (MIS). He is an Associate Professor and the Dean of the Faculty of Information Science and Technology. Before this position, he was a Research Fellow with United Nations University, a United Nations academic arm. He has conferred the Professional Technologist [.Ts/P.Tech.(IT)] credential from the Malaysian Board of Technology (MBoT), in 2018. During the Ph.D. study, he received the National Science Fellowship (NSF) Scholarship from the Malaysian Ministry of Science, Technology, and Innovation (MoSTI).

SHAYLA ISLAM (Senior Member, IEEE) received the B.Sc. degree in computer science and engineering from International Islamic University Chittagong, Bangladesh, and the M.Sc. and Ph.D. degrees from the Department of Electrical and Computer Engineering, International Islamic University Malaysia (IIUM), in 2012 and 2016, respectively. She is currently an Assistant Professor with UCSI University, Malaysia. She received the Malaysian International Scholarship for the Ph.D. study. She received the Silver Medal for her research work with International Islamic University Malaysia. In consequence, she also received the Young Scientist Award for the contribution of a research paper at the Second International Conference on Green Computing and Engineering Technologies 2016 (ICGCET'16), organized by the Department of Energy Technology, Aalborg University, Esbjerg, Denmark.

FATIMA RAYAN AWAD AHMED received the B.S. degree in computer science from the Sudan University of Science and Technology, in 2004, and the M.S. and Ph.D. degrees in computer science from Al Neelain University, Sudan, in 2007 and 2012, respectively.

She joined Prince Sattam Bin Abdulaziz University, Saudi Arabia, in 2013, as an Assistant Professor with the Information Systems Department, from 2013 to 2016, and has been with the Computer Science Department, since 2017. In 2004, she joined Sudanese Company for Telecommunications, Sudan, as a Computer Programmer with the IT Department, where she analyzed, designed, and programmed a set of systems. Her current research interests include artificial intelligence, systems and algorithms analysis and design, web applications, and e-learning.

AAISHANI DE is pursued the B.Tech. degree with the Department of Electronics and Communication Engineering, Indira Gandhi Delhi Technical University for Women. He is a Researcher. Her technology stack includes Java, Springboot and AI/ML development in python. She has published more than 10+ research articles in various international journals (including SCI, ESCI and Scopus indexed). Her research work primarily focuses on the topics of computer vision, natural language processing and deep learning. Her area of interests includes web development, machine learning, data science and artificial intelligence. She has received the Incentive Award for Excellence in Research from Indira Gandhi Delhi Technical University for her work.

MUHAMMAD ATTIQUE KHAN (Senior Member, IEEE) received the master's and Ph.D. degrees in human activity recognition for application of video surveillance and skin lesion classification using deep learning from COMSATS University Islamabad, Pakistan. He is currently a Lecturer with the Computer Science Department, HITEC University, Taxila, Pakistan. He has above 190 publications that have more than 6500 citations and impact factor more than 600 with H-index of 50. His research interests include medical imaging, COVID19, MRI analysis, video surveillance, human gait recognition, and agriculture plants. He is a Reviewer of several reputed journals, such as IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON NEURAL NETWORKS, *Pattern Recognition Letters*, *Multimedia Tools and Application*, *Computers and Electronics in Agriculture*, *IET Image Processing*, *Biomedical Signal Processing and Control*, *IET Computer Vision*, *EURASIP Journal of Image and Video Processing*, IEEE ACCESS, *Sensors* (MDPI), *Electronics* (MDPI), *Applied Sciences* (MDPI), *Diagnostics* (MDPI), and *Cancers* (MDPI).

TAHER M. GHAZAL (Senior Member, IEEE) received the B.Sc. degree in software engineering from Al Ain University, in 2011, the M.Sc. degree in information technology management from The British University in Dubai (associated with The University of Manchester and The University of Edinburgh), in 2013, the first Ph.D. degree in IT/software engineering from Damascus University, in 2019, and the second Ph.D. degree in information science and technology from Universiti Kebangsaan Malaysia, in 2023. With over a decade of extensive and diverse experience, he was an instructor, a tutor, a researcher, a teacher, a IT support/specialist engineer, and a business/systems analyst. He was with various departments, including engineering, computer science, and ICT. He was the Head of the STEM and Innovation, and has also been involved in quality assurance, accreditation, and data analysis, in several governmental and private educational institutions under KHDA, Ministry of Education, and the Ministry of Higher Education and Scientific Research, United Arab Emirates. He is actively involved in community services in the projects and research field. His research interests include the IoT, IT, artificial intelligence, information systems, software engineering, web development, building info, modeling, quality of education, management, big data, quality of software, and project management.