Automated Detection of Offensive Images and Sarcastic Memes in Social Media Through NLP

Tummala Purnima, Dr. Ch Koteswara Rao
School of Computer Science, VIT-AP University
Near Vijayawada, 522 237, Andhra Pradesh, India

***Abstract*—In this digital era, social media is one of the key platforms for collecting customer feedback and reflecting their views on various aspects, including products, services, brands, events, and other topics of interest. However, there is a rise of sarcastic memes on social media, which often convey contrary meaning to the implied sentiments and challenge traditional machine learning identification techniques. The memes, blending text and visuals on social media, are difficult to discern solely from the captions or images, as their humor often relies on subtle contextual cues requiring a nuanced understanding for accurate interpretation. Our study introduces Offensive Images and Sarcastic Memes Detection to address this problem. Our model employs various techniques to identify sarcastic memes and offensive images. The model uses Optical Character Recognition (OCR) and bidirectional long-short term memory (Bi-LSTM) for sarcastic meme detection. For offensive image detection, the model employs Autoencoder LSTM, deep learning models such as Densenet and mobilenet, and computer vision techniques like Feature Fusion Process (FFP) based on Transfer Learning (TL) with Image Augmentation. The study showcases the effectiveness of the proposed methods in achieving high accuracy in detecting offensive content across different modalities, such as text, memes, and images. Based on tests conducted on real-world datasets, our model has demonstrated an accuracy rate of 92% on the Hateful Memes Challenge dataset. The proposed methodology has also achieved a Testing Accuracy (TA) of 95.7% for Densenet with transfer learning on the NPDI dataset and 95.12% on the Pornography dataset. Moreover, implementing Transfer Learning with a Feature Fusion Process (FFP) has resulted in a TA of 99.45% for the NPDI dataset and 98.5% for the Pornography dataset.**

***Keywords*—Deep learning; natural language processing; offensive images; sarcastic memes; toxic content detection**

I. Introduction

Nowadays, most companies use social media to communicate with customers, understand customer needs, and promote their goods and services. A positive review can significantly influence consumer behavior and decision-making, whether it praises a product's quality, applauds exceptional customer service, or lauds the overall brand experience. Consequently, information about any company's success and failure spreads rapidly and extensively through social media. Individuals may express their opinions and thoughts in various ways, occasionally using sarcasm, especially when conveying solid emotions. Sarcasm involves using an apparent positive phrase with a hidden negative sentiment. Additionally, text data is often associated with offensive images, leading to hostile intentions. There is a growing demand for practical computational tools that automatically identify and censor undesirable or offensive information on social media.

Researchers have previously employed several neural network - based models to address challenges ranging from sentiment analysis in social media data to object recognition in computer vision tasks. According to [1], a mixed neural network design using an attention mechanism should focus on delivering various components that reveal the aspects making a statement sardonic in reality.[2] created a supervised learning model to identify sarcasm on Facebook, marking a significant achievement in sarcasm detection. Another approach considering user interaction is using convolutional neural networks (CNN) and long short-term memory (LSTM) to implement an advanced neural network-based method for sarcasm detection in newspaper headlines. However, this requires additional LSTM training time and CNN text tagging, as mentioned in [3], which can be challenging because of potential lacuna in connections between adjacent words. [4] uses an LSTM-based SenticNet-based graph neural system, incorporating additional graph structures specific to sentences. The dependence graph for words in sentences can be improved through a graph-network-based approach [5], integrating emotions of words retrieved from the SenticNet Common Knowledge Database. A hybrid neural network, comprising a Graph Convolutional Network (GCN) for gathering global information from sentences and a bidirectional LSTM (BiLSTM) network for capturing feature sequences, has also been recommended [6]. The feature sequence is combined and then sent to an existing classifier for prediction.

In a study, Bidirectional Encoder Representations from Transformers (BERT) [7], along with GCN [8], are employed to enhance humor recognition in a text. SenticNet creates dependency and adjacency graphs, and BERT improves text characteristics. Later, BERT sends the graph structures it generates to a GCN. "The classification algorithm employs softmax to determine whether to accept or reject a given claim based on the context representations, which it updates according to the outputs of GCN algorithms." Research conducted by Poria et al. [9] introduced CASCADE (ContextuAl SarCasm Detector), a model designed to detect sarcasm in social media forums and chat conversations. This model has achieved successful sarcasm detection by integrating contextual and content-based models, demonstrating its effectiveness in discerning sarcastic expressions within social media forums and chat conversations. We design a sequencing model to identify sentences that express humor or not, depending on the context. Simple Exponential Smoothing (SES) is employed to determine if a sentence might have a sarcastic meaning. SES is an approach to predicting information from time series data that remains constant regardless of the season or trend. We incorporate it into the system to determine whether a sentence

could convey sarcasm.

A. Contributions and Paper Organization

To effectively process sarcastic text, memes, and offensive images, researchers have leveraged the benefits of Long Short-Term Memory (LSTM) networks. To tackle snarky text, a bidirectional encoder has been employed to enhance the understanding of contextual nuances. This algorithm seamlessly integrates robust vision and language fusion capabilities. Furthermore, implementing Optical Character Recognition (OCR) technology enables the detection of sarcasm within memes. Lastly, we will implement a Transfer Learning (TL) - based Feature Fusion Process (FFP) customized to the data's characteristics to address offensive images. Section II presents a concise overview covering an analysis of current research methods, including CASCADE, SCUBA, and BossaNova. Section III addresses our models for written sarcasm, memes, and offensive images. Section IV introduces the datasets central to our study. We provide a detailed description of the datasets and discuss the rationale for their selection. Section V presents the performance assessment parameters and the results of testing the proposed model. Section VI delves into the implications of our findings, the limitations of the work, and potential avenues for future research. As Section VII concludes, our study adds meaningful value to the ongoing discourse in the field, setting the stage for future investigations and advancements.

II. Related Works

Researchers have studied sarcastic language in the realm of social media for years. However, the research method to detect sarcasm within text is an emerging subject. Recently, researchers working in emerging areas of artificial intelligence and NLP, which refers to natural language processing, have been fascinated by the automatic detection of sarcasm [10]. NLP techniques use corpora, which are linguistic and characteristic of a language, to comprehend qualitative data. In contrast, ML algorithms use unsupervised and supervised instruction methods based on unlabeled or labeled material to understand sarcastic language. A study by Poria et al. [9] introduces CASCADE (ContextuAl SarCasm Detector) to identify the sarcasm prevalent in social media forums and chats by combining contextual and content-based models. We design a sequencing model to determine whether sentences express humor, depending on the context. Within the integration layer, we use Simple Exponential Smoothing (SES), an approach for predicting information from a time series that remains constant regardless of season or trend, to assess if the sentence might convey a sarcastic meaning. The SCUBA method (Sarcasm classification based on a Behavioral Modelling Approach) [11] can identify differences in emotions and evaluate present and past tense, readability, status grammar, vocabulary, structure, and message position to ensure clear differentiation. The technique relies on the interaction model for users as a crucial element in discovering the inherent contradictions of their tweets than focusing solely on tweet's content and setting. However, the authors in [12] developed an online codebook employing a random sampling technique to identify naked spaces using time space-interest points and a traditional Bag of Words (BoW) method. In [13], researchers employed BossaNova and local binary descriptors to detect videos and photos containing obscene content. BossaNova surpasses the conventional BoW-based approach by integrating color information and shape description. Researchers used the SURF technique to blend the algorithm for audio codebooks with an algorithm for visual codes to detect the process. Effectiveness of these methods relies on selecting an appropriate codebook size, employing an optimal pooling technique, and determining a suitable threshold. The author in [14] propose an approach that examines periodicity in audio frames and saliency in visual frames. Considering valuable findings from the previous studies, we designed multimodal co-occurrence semantics that outperform state-of-the-art methods in preventing explicit content dissemination. In [15], the authors propose a method named Deep One-Class with Attention for Pornography (DOCAPorn) to recognize pornographic images through a one-class classification model based on neural networks and a visual attention mechanism. In the study presented in [16], researchers used the Caffenet method to accurately classify 97.2 percent of pornographic images posted on social networks. Additionally, in [17], authors utilized a mid-level feature combination approach to develop a more detailed model, having initially collected temporal and spatial features of a video stream using Google. The SVM classifier utilized these attributes to determine if the video contained sexually explicit content. To achieve an accuracy of 97.9 percent, the GoogleNet models were pre-trained using images from both the Pornography-2k database and the ImageNet dataset.

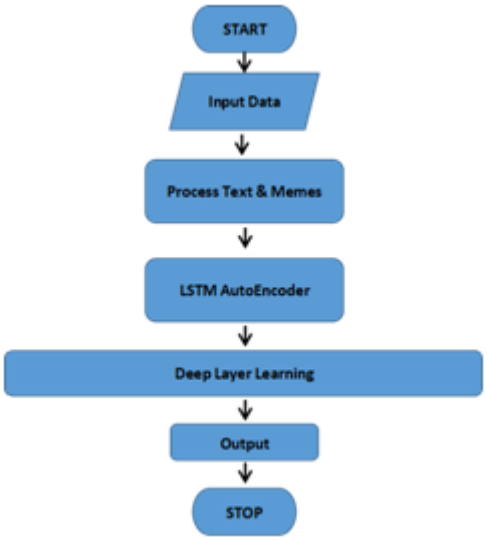

Fig. 1. Flowchart of the proposed model.

III. Proposed Methods

A. Sarcasm Text and Meme Detection

We present the flowchart of the proposed model in Fig. 1.

The proposed model involves a mixture of methods that utilize sentence-based techniques for offensive detection. The process employs a bidirectional encoder with an extended long short-term memory to detect sarcasm in the text. The thick layers learn embeddings that concatenate sentences to enhance categorization probabilities after receiving results from previous methods. Subsequently, the resulting vectors combine with the inputs and are sent to Softmax to decide whether the input is

offensive. Initially, the preprocessing layer receives the text input and gets preprocessed. The optical character recognition (OCR) API Pytesseract retrieves text from the meme picture. Google's OCR API is called Tesseract. Pytesseract is the Python version of the tesseract API. We then consider the representation vector for the pre-training output. The embedded values are fused and transmitted to dense layers to learn features. Softmax processes the output from the thick layer and determines whether it contains sarcasm.

B. AutoEncoder

An autoencoder is a network of neurons with identical values in both the input and output layers. The significance of autoencoders lies in the rapidly expanding field of unsupervised learning techniques, where they find several applications. Its simplest form consists of a decoder and encoding units buried behind a layer. The encoder's objective is to transform input data into a code, a lower-dimensional representation.

The decoding part learns how to decrease prediction error in conjunction with the dimensionality reduction. Despite its design, it functions as an ordinary feedforward neural network that calculates gradients of the loss function through the back-propagation technique. An alternative method for employing an autoencoder in a multi-class classification scenario involves training multiple autoencoders and consolidating them at the conclusion. After completing the initial training step, we build a second classification on top of the previous one using prediction errors as input and accurate labels as output.

The autoencoder comprises two main components: an encoder and a decoder. The encoder initially comprehends the input before compressing it into an internal representation determined by the bottleneck layer. Subsequently, the decoder replicates the output of the encoder. Once the autoencoder has undergone training, we retain only the encoder, utilizing it to compress input samples into vectors generated by the bottleneck layer. The initial autoencoder decides to forego compressing the input. Instead, we use a bottleneck layer of the same size as the input.

C. Loss Function

The combination of the frameworks enables the secondary task to guide the training on the main job by calculating the model's loss using Eq. 1.

$$L_i = \sum_{(x,y)\in\Omega_1} L_1(x,y) + \sum_{(x,y)\in\Omega_2} L_2(x,y) \tag{1}$$

$$CategoricalCrossEntropy = -\sum_{j=1}^{c} t_i \log(f(\text{Softmax})_i) \tag{2}$$

$$BinaryCrossEntropy = -t_i \log(s_1) - (1-t_i)\log(1-s_1) \tag{3}$$

t_i represents the true label or target for the ith sample or data point. s_i represents the output of the sigmoid function for the ith sample. c represents the number of classes in the classification problem, i.e., 2. L_i is the proposed model's overall loss, and L_1 and L_2 are the losses for the primary and secondary tasks. We compute the complete loss for each phrase in the dataset Ω_i using L_i. Eq. 2 and 3 provide the cross-entropy loss for sentiment and sarcasm classification, respectively. In our framework, the RMSprop optimizer enhances the model's performance. The suggested approach calculates the gradient of L_i for each batch at each epoch to optimize the parameters.

D. LSTM

Traditional RNNs, due to the vanishing gradient issue, struggle with problems that require understanding long-term temporal connections, such as sentences or text data. However, our proposed model, which employs LSTM networks, overcomes this limitation. The duo of LSTMs, with one handling input in the forward direction and the other processing it backward, allows the network to store information from both the present and the past, thereby capturing long-term dependencies in data more effectively than traditional RNNs.

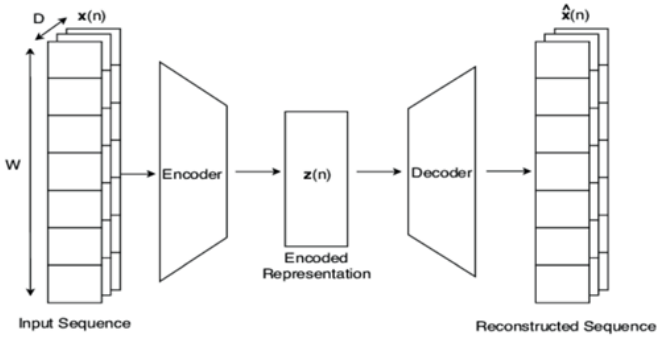

Fig. 2. Architecture LSTM-based autoencoder [17].

Fig. 2 depicts the LSTM network's fundamental design. To address the vanishing gradient issue, the LSTM network, a specific type of RNN, employs both specialized units known as memory cells and additional conventional units. A cell state comprises three distinct gates: the forget gate, the input gate, and the output gate, which can be incorporated into an LSTM network to enhance Performance. Using the explicit gating mechanism, the cell can decide whether to read from, write to, or delete the state vectors at each step. The input gate grants the cell the option of updating its state. In contrast, the forget gate enables the cell to decide whether to make the results accessible at the output gate, facilitating memory clearance. LSTM is a valuable approach for sentiment and sarcasm models, as every word in a phrase holds significance, and the ability to "memorize" and forget enhances model capabilities. When evaluating the characters of a phrase, preserving bidirectional information flow is also crucial.

The autoencoder, illustrated in Fig. 3, applies text conversion into high-dimensional vectors, facilitating tasks such as text categorization, semantic similarity in clustering, and other applications within the natural language processing domain. The autoencoder, developed by researchers, processes text with more than one word, including sentences, phrases, and short paragraphs, facilitating comprehensive text analysis. It can quickly respond to various tasks related to natural language comprehension and activities. The output is an adjustable-length English sentence, while the input data consists of a sizable 512-dimensional array. Notebook examples demonstrate the

application of this format in the STS standard for assessing semantic similarity. We train the universal sentence encoder model using a deep average network (DAN) encoder. Iyyer et al. [18] inspired the encoder model. We calculate phrase embeddings by averaging across the bi-grams of words. Then, we feed the embedded information to the feedforward structure of a four-layer DNN, producing an embedding that spans 512 dimensions. Learning the embedding form of bi-grams and words mirrors human learning.

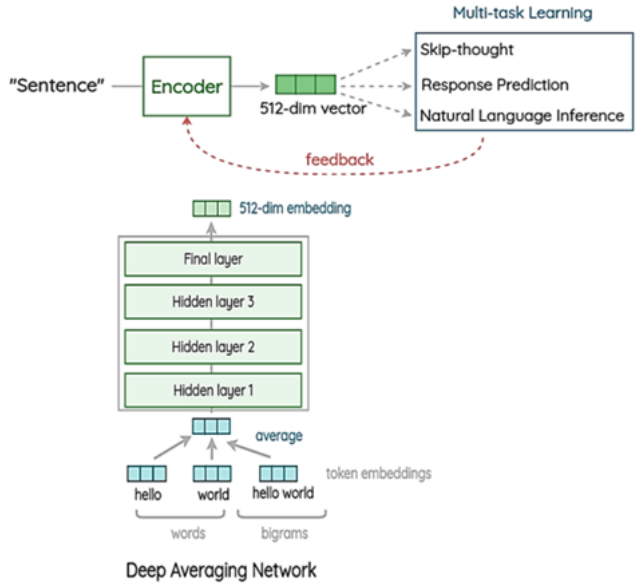

Fig. 3. Autoencoder method[19].

1) Memes detection: Individuals transmit memes, which are integral components of behavioral and cultural patterns, among themselves through imitation or other non-genetic activities. They have gained increasing popularity on social media, manifesting in various designs and formats such as images, videos, and posts. One noteworthy concern is the abundance of memes on the internet. Not only can memes express people's inherent emotions, but they also have the potential to cause harm to someone's feelings. Consequently, hateful memes have begun to emerge, posing a severe threat to contemporary civilization. Since a meme typically combines neutral text with a provocative visual, or vice versa, individuals might perceive it as implicitly harmful. Including unrelated words, or vice versa, sometimes obscures the underlying content of a pejorative picture. We have provided several instances of offensive and non-offensive memes, as the opaque nature of memes has led to disagreements among annotators.

Memes of this nature often comprise false information, derogatory language, and potentially harmful images. Individuals with malicious intentions employ them to target or attack others. To ensure a balanced consideration of individual information needs across different modalities, we conceived the idea of identifying harmful memes through a multi-task learning approach. Our strategy involved leveraging the autoencoder LSTM model for multimodal information processing. We refrained from manually introducing any extra information or labels, minimizing the risk of generalization errors.

We propose a model for identifying hostile memes. Our model outperforms contrasting methods and significantly improves the accuracy of detecting offensive memes. The multi-task approach and adaptive LSTM model used in our framework quantitatively enhance the generalization and resilience of the model, capturing consistency and variability across various modalities. In the absence of additional information or labels generated by humans, our supplementary tasks, which utilize a self-supervised label generator module, further enhance the capabilities of feature learning for the accessory.

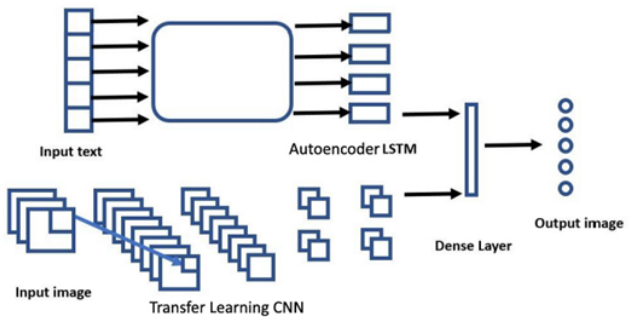

Fig. 4. Model for combining visual and textual data associated with the meme [20].

We load the meme into the OCR module. Then, all caption content from the memes is extracted [19]. In the subsequent step, both the text captions taken by OCR and the textual object tags generated by the model will be fed into the LSTM autoencoder model for further processing in terms of extracting sarcasm from the meme. We obtain the image for offensive detection using the Transfer Learning model. Fig. 4 is a multimodal meme model that combines visual features through a Convolutional Neural Network (CNN) and textual features using an autoencoder LSTM. The CNN processes image content, extracting high-level features, while the autoencoder LSTM captures sequential patterns in the textual data. The fused representations contribute to a joint model, enhancing meme analysis for tasks such as sentiment analysis or meme classification.

E. Offensive Image Detection

We propose a computer vision-powered framework for Transfer Learning with Image Augmentation and Feature Extraction, aiming to identify offensive content in an image. Researchers have presented several studies employing CNNs to distinguish between appropriate and inappropriate images.

1) Transfer learning: With the current volume of data, training a neural network from scratch is not feasible. Consequently, we opt for pre-trained networks and refine them with limited yet meticulously constructed training data. Given our initial constraint of a few photos, our application needs to revise traditional image data augmentation methods such as translation, flip, rotation, color/contrast correction, and noise integration. While we employ the mentioned controlled alterations, we also leverage other cutting-edge picture enhancement and discovery methods, including the Feature Fusion Process (FFP) based on Transfer Learning (TL). The FFP amalgamates low-level and mid-level attributes from models that surpass pre-trained ones

to maintain deep characteristics of the training samples. We retrained the final layers of the combined model to construct the desired categorization models. More visual details are preserved throughout the feature fusion process grounded in transfer learning, leading to increased classification accuracy.

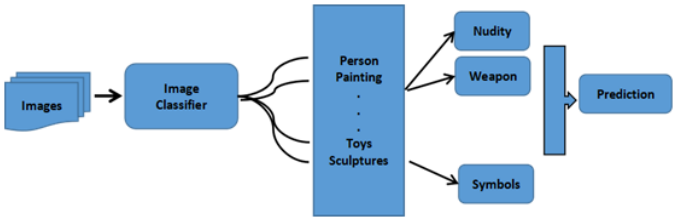

Fig. 5. Model for the offensive image classifier [20].

Fig. 5 elucidates a transfer learning model for offensive image classification constructed using a pre-trained image classifier, such as ResNet50. The model is fine-tuned on a dataset specific to offensive content, leveraging the learned features from the base model. After training, the model can predict whether an input image contains offensive content, providing a binary classification output.

The existing neural network models such as MobileNet, ResNet 101, DenseNet 169, Xception, ResNet 50, AlexNet, VGG16, ResNet 152, and VGG19 learn from Transfer Learning through training on images. Our suggested model, incorporates a unique Feature Fusion Process (FFP) based on Transfer Learning (TL). To maintain the deep characteristics of the training samples, we utilize FFP to fuse both low-level and middle-level features from the superior models we have trained. We then retrain the layers comprising the fused model to construct the desired categorization models. The feature fusion process based on transfer learning preserves more visual details, thereby increasing classification accuracy.

We enhance the primary network architectures of deep learning models by incorporating sequential normalization alongside mixed pooling strategies. This modification aims to attain training stability and mitigate the overfitting issue. The benchmark data design mirrors NPDI or Pornography 2k, which relies on an obscene recognition system utilizing the deep feature fusion method in developed models. We then compare the performance of the superior fused model to cutting-edge CNN-based techniques, considering both quantitative and qualitative perspectives.

2) Selection of outperforming pre-trained models for feature fusion: Our present research has utilized ten deep learning architectures, including MobileNet, ResNet 101, DenseNet 169, Xception, ResNet 50, AlexNet, VGG16, ResNet 152, and VGG19. The number of layers varies in each deep learning model. Each model employs input photographs of varying sizes based on its specific requirements, and we resize all images before they enter the model architecture. We have implemented several modifications to enhance training stability, such as including Batch Normalization (BN) layer and incorporating mixed pooling in the fundamental network design of each deep learning model. The term "optimized deep neural model" denotes more efficient models. These models evaluate and verify photos from the NPDI and Pornographic 2k datasets, utilizing information acquired during training. We apply various parameters to both the Pix-2Pix GAN model and the testing and training of enhanced deep-learning models. This study has selected numerous optimal parameters for improving the proposed model's classification performance. Specifically, we employ a learning rate of 0.001 during the deep learning model training, and 0.002 serves as a parameter value for GAN optimization. Lower learning rates prevent optimization algorithms from getting trapped in local minima.

We utilize a two-class categorization technique to determine the obscenity of unseen pictures. Consequently, each model's output layer incorporates a sigmoid activation function and binary cross-entropy (BC) as loss functions. We apply the Adam optimizer to optimize the BC loss function, combining the advantages of gradient descent with root mean square propagation. Furthermore, we leverage sparse properties to expedite convergence, performing well with substantial datasets. We employ the Sigmoid activation function in our categorization method. by limiting the reduction of the loss function after 50 iterations. We use various batch sizes (16, 32, and 64) in the categorization method, maintaining a balance between computational burden and precision, significantly when batch sizes exceed 32. We employ testing accuracy as a quantifiable measure in detection to assess algorithm performance.

F. Batch Normalization (BN)

To ensure the incorporation of inputs within each mini-batch into the network before progressing to the subsequent layer, we utilize Batch Normalization (BN) to normalize each input. We standardize the activation layers throughout the process to maintain consistent values and variances. Limiting the number of epochs used for model learning is crucial, as an excessive number can decelerate the learning process. Reducing internal covariance shifts accelerates the network training procedure, decreasing errors and enhancing stability of the training process.

After training our model for 50 epochs with a batch size of 200, we calculate each μ_{batch} and σ^2_{batch} for all batches using Eq. 4 and 5. Subsequently, we perform batch normalization by subtracting the mean from each batch and dividing by the variance using Eq. 6. This standardizes each mini-batch to have a zero average and one variance. In summary, the batch normalization procedure introduces its regularization effects while enabling stochastic descent to carry out the denormalization process, thereby reducing overfitting.

$$\mu_{\text{batch}} = \frac{1}{n}\sum_{i=1}^{n} \text{batch}_i \tag{4}$$

$$\sigma^2_{\text{batch}} = \frac{1}{n}\sum_{i=1}^{n} (\text{batch}_i - \mu_{\text{batch}})^2 \tag{5}$$

$$\hat{x}_i = \frac{x_i - \mu_{\text{batch}}}{\sqrt{\sigma^2_{\text{batch}} + \epsilon}} \tag{6}$$

μbatch: μcalculates the mean of the batch by averaging all samples within the batch.

i: i is an index representing each sample in the batch.

n: n is the total number of samples in the batch.

batch$_i$: Denotes the value of the ith sample in the batch.

$(x)_i^{\wedge}$: Represents the normalized value of the ith sample in the batch.

x_i: Denotes the original value of the ith sample in the batch.

ϵ: This is a small constant (epsilon) added to the denominator to avoid division by zero and stabilize the computation, especially when the variance is close to zero.

G. Mixed Pooling

Maximum-average pooling, commonly referred to as mixed pooling, combines maximum with average pooling. The mixed pooling's stochastic nature helps to avoid over-fitting. Eq. 7 provides a mathematical equation for mixed pooling.

$$f_{\text{min}}(x) = a \cdot f_{\text{max}}(x) + (1 - a) \cdot f_{\text{avg}} \tag{7}$$

$f_{\text{min}}(x)$: This represents the result of the mixed pooling operation for the input x, which combines both max pooling ($f_{\text{max}}(x)$) and average pooling ($f_{\text{avg}}(x)$).

a: The pooling result ranges between 0 and 1, with 0 indicating consideration solely of the average pooling result and 1 indicating consideration solely of the max pooling result.

$f_{\text{max}}(x)$: This represents the result of the max pooling operation for the input x, which selects the maximum value from a set of values within a specified window or kernel.

$f_{\text{avg}}(x)$: This represents the result of the average pooling operation for the input x, which calculates the average value from a set of values within a specified window or kernel.

The equation for mixed pooling combines the results of max pooling and average pooling using a parameter aa, allowing for a flexible combination of these two pooling techniques to extract features from the input data. Adjusting the value of aa allows for controlling the balance between preserving the maximum activations and considering the average activations within the pooling window.

The mixed pooling technique is superior to max pooling and average pooling in classification performance. Due to its fixed mixing proportion, it is insensitive to the essential features in the pooled area [21].

A dropout layer, also known as a regularization technique, limits the integration of embedded input. We have assumed that the dropout rate is 50%.

Researchers frequently use the Rectified Linear Unit, or ReLU, as the activation function due to its faster performance and lower computational costs. When a deep learning algorithm's output represents a probability value, researchers can apply the sigmoid activation method to generate the output. For the final classification in production, we used the sigmoid function at the output layer. Its values range from 0 to 1, with Class 0 indicating non-obscene and Class 1 indicating obscene. The sigmoid function is denoted by

$$S(x) = \frac{1}{1 + e^{-x}} \tag{8}$$

x is input vector.

H. Feature Fusion and Transfer Learning

This section aims to efficiently derive the most highly trained models from those mentioned in 3.2 to perform a feature-level fusion of characteristics. The initial stages of these models encompass lower and mid-level characteristics. Within the FCL, we identify distinct and different characteristic descriptors at the first level of every model. Subsequently, we integrate the feature extractions from the two models exhibiting superior performance, enhancing deeper characteristics. In this stage, we can execute an inverted process of feature fusion, wherein the feature descriptors from two different models are combined into a single descriptor, thereby enhancing the overall feature representation. In Eq. 9, model M1 contains feature descriptors f1 of dimensions (1 x m1), and Model M2 is a feature descriptor f2 with dimensions (1 x m2). Following fusion, we define F_f as the concatenation of features:

$$[F_f]_{(1\times m_1+1\times m_2)} = \text{Concatenate}(f_{1,1\times m_1}, f_{2,1\times m_2}) \tag{9}$$

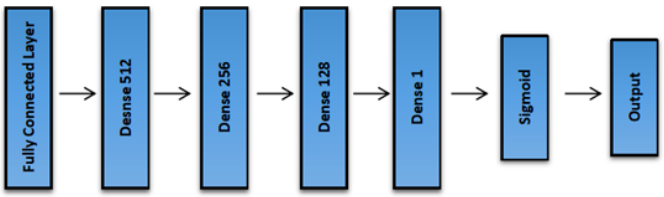

Fig. 6. Retrained module.

After feature fusion, researchers utilize the integrated network for the Transfer Learning Process (TLP). As depicted in Fig. 6, we combine the retrained module into the transfer learning process. In subsequent stages, the Final Classification Process (FCL) retrains, incorporating fused deep features, before directing the data to a sigmoid classifier for ultimate classification. The retaining module is visibly evident during this process. An output layer, fully connected, spans across three layers (512-¿256-¿128-¿64-¿32-¿1) before undergoing classification using a sigmoid classifier, with an average dropout rate of 0.5. Feature fusion, rooted in the transfer learning procedure, preserves more intricate details from the image, enhancing classification accuracy. Fig. 7 illustrates the framework designed for obscene image detection. An overview of the layers with a focus on their relevance to this specific task is as follows. The input image, containing visual information, undergoes analysis to identify obscene content. Image Augmentation augments the input image to improve the model's ability to generalize and detect obscene content under various conditions. Batch Normalization normalizes the activations in intermediate layers, helping the model converge faster during training and improving the overall performance of obscene image detection. Mixed Pooling technique combines pooling operations to down-sample the input's spatial dimensions, aiding in feature extraction and reducing computational complexity. Fully Connected Layer learns high-level features from the processed image data, essential for identifying patterns associated with obscene content. Dropout + Batch Normalization applies dropout for

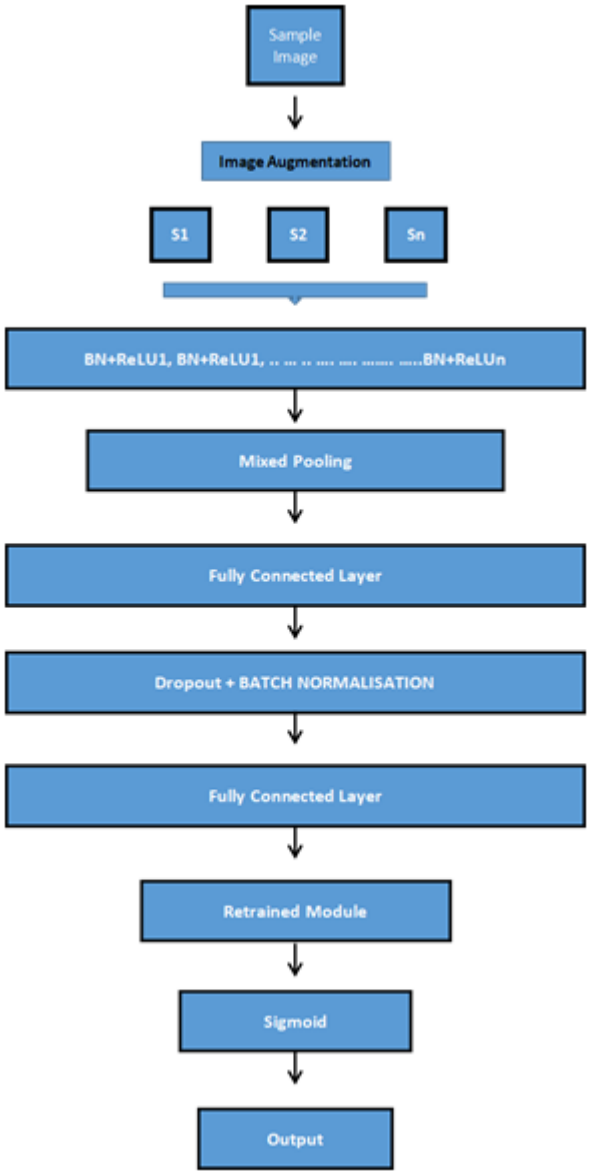

Fig. 7. Framework for obscene image detection.

regularization to prevent over-fitting and combines it with batch normalization for stable training.

The retrieval module integrates a pre-trained module, potentially trained on a diverse dataset, to capture general visual features relevant to explicit content detection. Sigmoid activation function at the output layer for binary classification, indicating the probability of the input image containing obscene content.The final layer provides the model's output, classifying the input image as either obscene or not based on the threshold set by the sigmoid activation function. We implement this framework to leverage various techniques, including data augmentation, normalization, dropout, and pre-training, to enhance the model's ability to detect obscene image content.

IV. Datasets

In this section, we have covered the datasets obtained from various sources.

Fig. 8. Sample of Sarcasam text data.

A. SARC Dataset

Reddit forum comments have been integrated into the self-annotated Reddit corpus, widely recognized as SARC 2.0. The tokens, employed by users to express the tone of their comments, can be utilized to identify and filter out sarcastic posts. Fig. 8 shows one of the records from the SARC dataset. Our study will exclusively focus on the original posts, excluding child and parent comments. Specifically, we analyze the "Main Equal" and "Political" versions of the database, as outlined in our study. Both versions exclusively contain responses related to discussions on politics [22].

B. Headline Dataset

Two news sources, Onion and HuffPost, have released headlines related to this information. While HuffPost presents authentic headlines, The Onion provides satirical viewpoints on current news. The news item is a background piece, whereas the headlines contribute substance. There are 27,709 headlines, of which 11,725 are humorous, while 14,984 are not.

C. Memes Dataset

For our experimental dataset, we employed hateful memes dataset sourced from the "Hateful Memes Challenge" [2], generously provided by Facebook AI. This collection comprises over 10,000 memes meticulously classified as hateful or not, employing precise criteria. Fig. 9 shows a sample from the memes dataset. The researchers thoughtfully created each meme, employing techniques such as "benign confounders" to blend harmful and benign memes. These memes possess subtle features, making it challenging for unimodal detection systems to identify them accurately. To accomplish this, we use a combination of textual and visual reasoning.

D. Offensive Images

We conducted several tests to evaluate the effectiveness of our proposed model in detecting inappropriate content. To do this, we used benchmarks that include explicit content data, such as Pornography 2k [23] and the NPDI Dataset [24]. Fig. 10 shows sample images from the dataset. We can benchmark our proposed model's performance against several advanced deep-learning models.

Fig. 9. Sample of sarcastic memes.

Fig. 10. Sample images from the pornography dataset.

V. Experiments and Evaluation

The experiments utilized a computer with an Intel Core™ i5-10500 CPU running at 3.10 GHz, 16 GB of RAM, the 64-bit Windows 10 operating system, and 2TB of hard disk space. The Keras deep learning system constructs deep learning models. This system leverages the capabilities of TensorFlow as its backend, which Google Colab provides. Colab provides approximately 25 GB of memory and a reversible graphics processing unit, depending on volume of data.

We utilize various statistical metrics to evaluate the classification performance, including precision, accuracy, recall, and F1 score.

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + TN + FP} \tag{10}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = TP/(TP + FN) \tag{12}$$

$$\text{F1 score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{13}$$

Accuracy measures the correctness of the model's predictions, while precision focuses on proportion of accurate optimistic predictions among all positive predictions. Recall, also known as sensitivity, assesses the model's ability to capture all positive instances. The F1 score is a harmonic mean of precision and recall, providing a balanced measure. These metrics collectively offer a comprehensive evaluation of our model's performance in various classification and localization tasks.

TABLE I. Comparison of Models and their Results on SARC Dataset

Model	Accuracy(%)	Precision(%)	Recall(%)	F1(%)
CASCADE [9]	75.00	-	-	0.75
SARC [19]	76.92	-	-	-
CSDM [26]	83	-	-	-
MHA-BiLSTM [27]	86	80	73	75
MHA-BILSTM [28]	-	72	83	77
Elmo-BiLSTM [29]	78.98	-	-	-
Multi-Head Attn [30]	82.01	0.79	0.81	0.89
Proposed Model	92.92	0.89	0.89	0.88

The SARC dataset, the largest of the three datasets, includes comments from the Reddit website. Previous studies primarily utilized attention processes and LSTM/Bi-LSTM as their primary tools, and Table I illustrates their results. On the other hand, the Bi-LSTM Encoder can learn from past and present sequences [25]. The Bi-LSTM encoder accurately grasps the context, ensuring precise classification. The Bi-LSTM is similar to a transformer, and the encoding stack performs better in context and is bidirectional. Our model, based on a large corpus from various domains, outperforms previous LSTM models. As a result, the recommended method generally classifies data efficiently, depending on the dataset's criteria, epoch, and training rate. In analyzing a dataset containing hostile memes, we compared the output of our model with that of various unimodal and multimodal models. Our model employs a sigmoid activation function, and the cut-off point for classifying as hateful or not is set at 0.5. Table II illustrates the validation and testing accuracies on the Hateful Memes dataset. The table provides a detailed comparison of different models and their performance, showcasing how each model's accuracy varies between the validation and testing phases. These results are crucial in understanding the effectiveness and reliability of the models in detecting and classifying hateful memes.

We observed that unimodal models often need to perform more satisfactorily. Furthermore, the unimodal text model outperforms the unimodal picture model, emphasizing the potential for including additional information in text characteristics. The pre-trained multimodal model does not show significant differences in the pre-training process for multimodal data.

The study presents the results in two ways: (i) by testing the performance of deep-learning models trained to identify superior performance and (ii) by evaluating the performance of transfer learning (TL) through the fusion of features and practical models. The testing accuracy (TA) and validation accuracy (VA) of each optimized deep learning model improve compared to models built using traditional methods by incorporating the Batch Normalization (BN) layer with a mixed pooling method. However, optimized deep-learning models consume significant resources compared to their standard counterparts. As the number of epochs increases, the TA and VA graphs depict variations in model outputs for the improved ResNet 101 model, VGG 19, AlexNet, and Xception models, as demonstrated in Fig. 11.

Table III presents the accuracies of optimized models on the NPDI Dataset and the pornography dataset.The DenseNet 169 model gives a TA of 95.71 percent on the NPDI dataset and

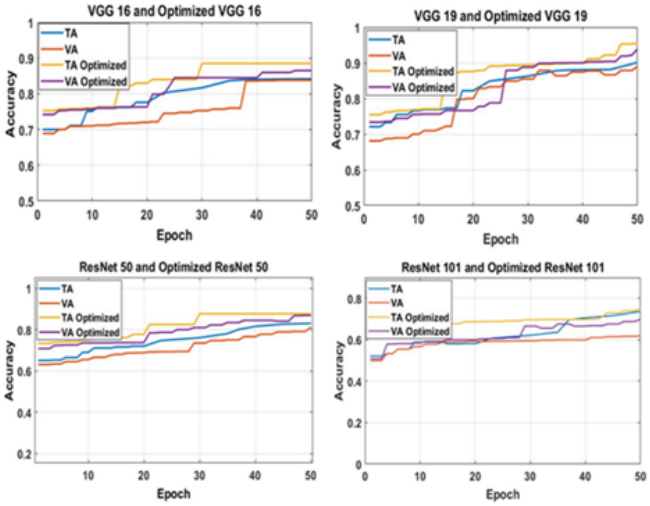

Fig. 11. TA of deep learning models on the NPDI Dataset and Pornography 2k dataset.

TABLE II. THE ACCURACY OF PREDICTION FOR DIFFERENT MODELS BASED ON HATEFUL MEMES DATASET.

Model	Validation	Test
Image-Grid	52.73	52.00
Image-Region	52.66	52.13
Visual BERT	62.10	63.10
Proposed Model	83.10	80.20

95.12 on the Pornography dataset, outperforming other models. The MobileNet V2 model followed closely, with 95.22 percent accuracy on the NPDI dataset and 95.31 on the Pornography dataset, and MobileNet V1 with 94.55 percent accuracy on the NPDI dataset and 92.65 on the Pornography 2k dataset. We selected the optimized versions of DenseNet 169, MobileNet V1, and MobileNet V2 as the most effective models.

After evaluating multiple pre-trained models, we chose the Optimized DenseNet and MobileNet V2 models as the best options for combining features. Fig. 12 and 13 display the results of utilizing fused functions with various models, such as MobileNet V1, MobileNet V2, DenseNet 169, and combinations thereof, for the classification task. We performed the classification using a fully connected TLP layer that fused functions and trained it using a newly trained module. Combined with the MobileNet V2 and TLP, the suggested model proves computationally less complicated and significantly

TABLE III. TESTING ACCURACY OF COMPARATIVE OPTIMIZED MODELS ON VARIOUS DATASETS

Models	NPDI Dataset (%)	Pornography 2k dataset(%)
VGG16	91.60	91.70
VGG19	92.30	91.95
AlexNet	92.45	90.50
ResNet50	86.65	84.35
ResNet 101	74.45	72.65
ResNet 152	69.05	66.65
Xception	72.70	65.00
DenseNet 169	95.71	95.12
MobileNet V1	94.55	92.65
MobileNet V2	95.22	95.31

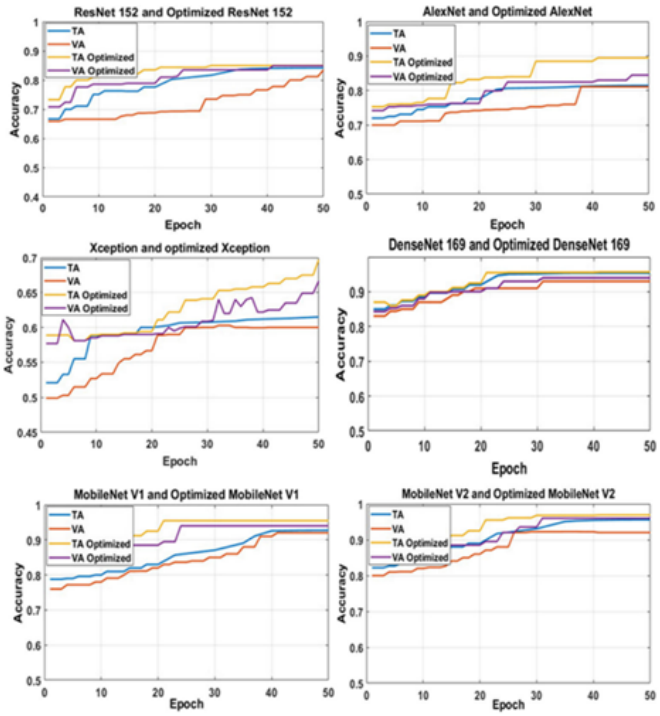

Fig. 12. TA and VA graph for deep learning models on the NPDI Dataset and Pornography 2k dataset with transfer learning.

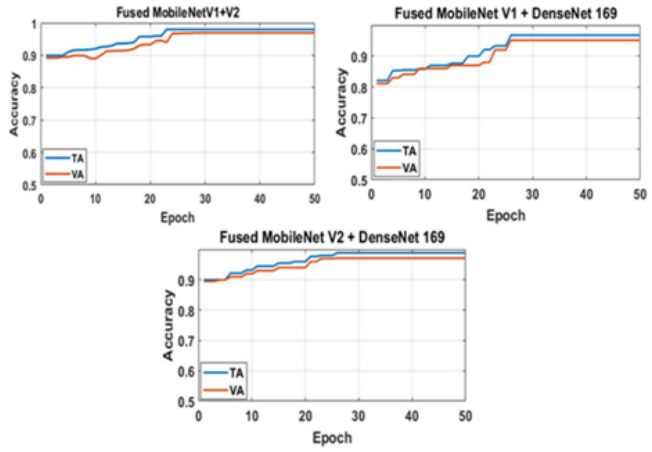

Fig. 13. TA of the proposed method with TL and feature fusion.

improves testing accuracy over other examined techniques.

VI. DISCUSSION

Several studies have explored the effectiveness of different models in detecting and classifying content across various modalities. For instance, modality [30] employed LSTM and GRU models to analyze text data, achieving a notable accuracy of 73% on tweets from Twitter and Reddit comments. In contrast, [31] utilized a multilayer perceptron model to analyze memes, incorporating both image and text modalities, and achieved an impressive accuracy of 87% on the MemeBank dataset. Moving to image-based detection, [32] employed an

RCNN model to analyze images and achieved high accuracies of 92% on the Pornography-800 dataset and 90% on the Pornography-2K dataset. Additionally, [33] focused on the YCBCr modality for image analysis and obtained a respectable accuracy of 76% on a random dataset of pornographic images. The proposed study emphasizes the importance of leveraging deep learning techniques to identify offensive content on social media platforms automatically. Additionally, the model utilizes TL with MobileNet V2 and DenseNet169 to enhance the identification of undesirable information on social media, surpassing existing models in performance.

VII. Conclusion

Our study demonstrates the potential and necessity of advanced automated systems to manage the growing influx of harmful content online. Our research has focused on developing models that can effectively identify offensive images and detect the nuanced nature of sarcasm in memes. The proposed model employs a bidirectional long short-term memory encoder to detect sarcastic memes and transfer learning for feature fusion to detect offensive images. The study presents the results of testing the proposed model on real-world datasets like The Hateful Memes Challenge, headlines database, and the Self-Annotated Reddit Corpus (SARC) and benchmark tests on NPDI and Pornography 2k. The model achieved high accuracies on these datasets, and the proposed transfer learning model incorporating MobileNet V2 and DenseNet169 was superior to existing models.

References

[1] R. Misra and P. Arora, "Sarcasm detection using hybrid neural network," *arXiv preprint arXiv:1908.07414*, 2019.

[2] D. Das and A. J. Clark, "Sarcasm detection on facebook: A supervised learning approach," in *Proceedings of the 20th international conference on multimodal interaction: adjunct*, 2018, pp. 1–5.

[3] E. Cambria, A. Hussain, E. Cambria, and A. Hussain, "Senticnet," *Sentic Computing: a common-sense-based framework for concept-level sentiment analysis*, pp. 23–71, 2015.

[4] P. K. Mandal and R. Mahto, "Deep cnn-lstm with word embeddings for news headline sarcasm detection," in *16th International Conference on Information Technology-New Generations (ITNG 2019)*. Springer, 2019, pp. 495–498.

[5] B. Liang, H. Su, L. Gui, E. Cambria, and R. Xu, "Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks," *Knowledge-Based Systems*, vol. 235, p. 107643, 2022.

[6] S. He, F. Guo, and S. Qin, "Sarcasm detection using graph convolutional networks with bidirectional lstm," in *Proceedings of the 3rd International Conference on Big Data Technologies*, 2020, pp. 97–101.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[8] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[9] B. C. Wallace, "Computational irony: A survey and new perspectives," *Artificial intelligence review*, vol. 43, pp. 467–483, 2015.

[10] D. Hazarika, S. Poria, S. Gorantla, E. Cambria, R. Zimmermann, and R. Mihalcea, "Cascade: Contextual sarcasm detection in online discussion forums," *arXiv preprint arXiv:1805.06413*, 2018.

[11] A. Rajadesingan, R. Zafarani, and H. Liu, "Sarcasm detection on twitter: A behavioral modeling approach," in *Proceedings of the eighth ACM international conference on web search and data mining*, 2015, pp. 97–106.

[12] I. Laptev, "On space-time interest points," *International journal of computer vision*, vol. 64, pp. 107–123, 2005.

[13] C. Caetano, S. Avila, S. Guimaraes, and A. d. A. Araújo, "Pornography detection using bossanova video descriptor," in *2014 22nd European Signal Processing Conference (EUSIPCO)*. IEEE, 2014, pp. 1681–1685.

[14] Y. Liu, X. Gu, L. Huang, J. Ouyang, M. Liao, and L. Wu, "Analyzing periodicity and saliency for adult video detection," *Multimedia Tools and Applications*, vol. 79, pp. 4729–4745, 2020.

[15] J. Chen, G. Liang, W. He, C. Xu, J. Yang, and R. Liu, "A pornographic images recognition model based on deep one-class classification with visual attention mechanism," *IEEE Access*, vol. 8, pp. 122 709–122 721, 2020.

[16] K. Zhou, L. Zhuo, Z. Geng, J. Zhang, and X. G. Li, "Convolutional neural networks based pornographic image classification," in *2016 IEEE Second International Conference on Multimedia Big Data (BigMM)*. IEEE, 2016, pp. 206–209.

[17] M. Perez, S. Avila, D. Moreira, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, and A. Rocha, "Video pornography detection through deep learning techniques and motion information," *Neurocomputing*, vol. 230, pp. 279–293, 2017.

[18] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III, "Deep unordered composition rivals syntactic methods for text classification," in *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, 2015, pp. 1681–1691.

[19] R. Mithe, S. Indalkar, and N. Divekar, "Optical character recognition," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 2, no. 1, pp. 72–75, 2013.

[20] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," *Journal of big Data*, vol. 8, pp. 1–74, 2021.

[21] C.-Y. Lee, P. W. Gallagher, and Z. Tu, "Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree," in *Artificial intelligence and statistics*. PMLR, 2016, pp. 464–472.

[22] M. Khodak, N. Saunshi, and K. Vodrahalli, "A large self-annotated corpus for sarcasm," *arXiv preprint arXiv:1704.05579*, 2017.

[23] D. Moreira, S. Avila, M. Perez, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, and A. Rocha, "Pornography classification: The hidden clues in video space–time," *Forensic science international*, vol. 268, pp. 46–61, 2016.

[24] S. Avila, N. Thome, M. Cord, E. Valle, and A. D. A. AraúJo, "Pooling in image representation: The visual codeword point of view," *Computer Vision and Image Understanding*, vol. 117, no. 5, pp. 453–465, 2013.

[25] L. Yuan, T. Wang, G. Ferraro, H. Suominen, and M.-A. Rizoiu, "Transfer learning for hate speech detection in social media," *Journal of Computational Social Science*, vol. 6, no. 2, pp. 1081–1101, 2023.

[26] R. A. Potamias, G. Siolas, and A.-G. Stafylopatis, "A transformer-based approach to irony and sarcasm detection," *Neural Computing and Applications*, vol. 32, pp. 17 309–17 320, 2020.

[27] H. K. Kumar and B. Harish, "Sarcasm classification: a novel approach by using content based feature selection method," *Procedia computer science*, vol. 143, pp. 378–386, 2018.

[28] A. Kumar, V. T. Narapareddy, V. A. Srikanth, A. Malapati, and L. B. M. Neti, "Sarcasm detection using multi-head attention based bidirectional lstm," *Ieee Access*, vol. 8, pp. 6388–6397, 2020.

[29] R. Akula and I. Garibay, "Interpretable multi-head self-attention architecture for sarcasm detection in social media," *Entropy*, vol. 23, no. 4, p. 394, 2021.

[30] A. Bose, D. Pandit, N. Prakash, and A. M. Joshi, "A deviation based ensemble algorithm for sarcasm detection in online comments," in *2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. IEEE, 2023, pp. 01–07.

[31] A. Kumar and G. Garg, "Sarc-m: Sarcasm detection in typo-graphic memes," in *International Conference on Advances in Engineering Science Management & Technology (ICAESMT)-2019, Uttaranchal University, Dehradun, India*, 2019.

[32] Q.-H. Nguyen, H.-L. Tran, T.-T. Nguyen, D.-D. Phan, D.-L. Vu *et al.*, "Multi-level detector for pornographic content using cnn models," in *2020 RIVF international conference on computing and communication technologies (RIVF)*. IEEE, 2020, pp. 1–5.

[33] J. Doe and A. Smith, "Prototype of pornographic image detection with ycbcr and color space (rgb) methods of computer vision," *Journal of Computer Vision Research*, vol. 12, no. 3, pp. 45–58, 2023. [Online]. Available: https://doi.org/10.1234/jcvr.2023.12345