Proceedings of the 2nd International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/76/20240603

Detecting unfriendly memes: A machine learning approach for content moderation

Ziao Huang

Faculty of Science and Engineering, University of Nottingham Ningbo China, Ningbo, 315000, China

corresponding author: smyzh4@nottingham.edu.cn

Abstract. The article presents a machine learning approach to address concerns about the dissemination of unfriendly content through memes. Specifically, it focuses on detecting unfriendly memes for content moderation purposes. It will compare three different methods which respectively focusing on textual content, image content and the combination of text and image content. The approach utilizes advanced machine learning models to distinguish between friendly and unfriendly content by leveraging both textual and visual elements of memes. State-of-the-art techniques such as Optical Character Recognition (OCR) for text extraction and Convolutional Neural Networks (CNNs) for image analysis are employed. Additionally, recurrent neural networks (RNN), gated recurrent units (GRU), and long short-term memory (LSTM) models are utilized for text classification. MLP is used in the classification process of combination. The models' performance is evaluated using objective measurements such as accuracy, ROC curves, and confusion matrices. The results demonstrate the effectiveness of the proposed approach in identifying unfriendly content and its implications for content detection on social media platforms.

Keywords: Unfriendly memes, machine learning algorithms, content moderation.

1. Introduction

In recent years, the proliferation of memes as a common form of online communication has raised concerns about the dissemination of unfriendly content. Automated systems for evaluating the offensiveness of memes on social media platforms are essential for efficiently and effectively monitoring a vast amount of content, ensuring timely detection and response to offensive material [1]. Unfriendly memes, including those containing hate speech and misinformation, contribute to toxic online environments and pose challenges for content moderation systems [2, 3]. Addressing this issue requires effective detection mechanisms capable of identifying and reduce harmfulness in real-time [4].

Machine learning techniques offer promising avenues for detecting unfriendly memes [5, 6]. Memes typically consist of both textual and visual elements. Firstly, the textual content within the memes can be identified through Optical Character Recognition (OCR). By utilizing Natural Language Processing (NLP) algorithms to analyze textual content and computer vision algorithms to interpret visual elements, it is possible to develop robust detection models capable of distinguishing between friendly and unfriendly memes [7]. For instance, a study employed Multi-Layer Perceptron (MLP) to detect hate speech in memes by analyzing both visual and language representations [8].

© 2024 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/76/20240603

In another study on multimodal religiously hateful social media memes, Stacked LSTM and BiLSTM models were utilized for text classification [9]. Convolutional Neural Networks (CNN) are often used to categorize images into predefined classes based on both image and textual features [10]. Additionally, object detection algorithms can identify and localize specific objects or regions of interest within an image, facilitating tasks such as face recognition and scene understanding [11].

This paper proposes a comprehensive approach to using machine learning, incorporating text and image recognition algorithms, for unfriendly meme detection. The methodology will be outlined, including data preprocessing, feature extraction, model selection, training and evaluation. Through empirical analysis, this paper demonstrates the effectiveness of the approach in identifying unfriendly content and discuss its implications for content moderation on social media platforms.

2. Methodology

In the methodology section, a multi-model approach was adopted to address image classification, text classification, and combined image-text classification tasks. After preprocessing the data, feature extraction was performed separately from textual and visual content respectively, followed by classification using various models. Evaluation of different models' performances was conducted. Subsequently, integration of the textual and visual methods was implemented to combine their classification capabilities.

2.1. Data Preprocessing

The data is taken from The Facebook Hateful Meme Dataset and the Hate Speech and Offensive Language Dataset due to their comprehensive coverage of social media content, including memes, speech, and textual data [12, 13].

The Facebook Hateful Meme Dataset comprises a diverse collection of memes extracted from Facebook, annotated for hate speech and offensive content. Similarly, the Hate Speech and Offensive Language Dataset contain textual data from various online platforms, annotated for hate speech and offensive language. These datasets provide a rich source of unfriendly content for training and evaluation purposes.

Before feature extraction and model training, the datasets underwent several preprocessing steps to ensure consistency and quality. Textual preprocessing included tokenization, stemming, and lemmatization to standardize the textual content across memes. Additionally, stop words were removed to reduce noise in the textual data. For image preprocessing, images were resized to a standard resolution and converted to grayscale to simplify analysis. Irrelevant or noisy visual elements were removed to focus solely on the essential features relevant to meme classification.

2.2. Feature Extraction

Optical Character Recognition (OCR) was used to recognize textual information from memes and derived textual features using techniques such as word embedding. This paper also utilized Natural Language Processing (NLP) algorithms to analyze textual content. The popular word embedding method, GloVe, was utilized to extract textual features. GloVe generates word vectors by incorporating global statistical information, which captures semantic relationships between words. By incorporating GloVe embeddings, textual features extracted from meme captions or associated text better represent the semantic meanings of words and phrases. This improves feature representation, enhancing machine learning models' ability to understand and process textual information in meme content. This contributes to improved performance in various tasks on meme datasets.

Visual features, on the other hand, were extracted using a VGG-based Convolutional Neural Network (CNN), which is a class of deep learning models suitable for analyzing image data. CNN automatically learns hierarchical representations of visual features, enabling it to capture complex patterns and structures within images. Relevant visual features were extracted from the memes' images using pre-trained CNN models or custom CNN architectures, providing additional context for model training and classification.

DOI: 10.54254/2755-2721/76/20240603

2.3. Model Selection

For image classification, the methodology began with feature extraction using the VGG architecture. Subsequently, classification was performed using Random Forest, Support Vector Machine (SVM), and a custom Convolutional Neural Network (CNN). The custom CNN was optimized using the Adam Optimizer to enhance its performance. The evaluation of performance included metrics such as accuracy, which was summarized in a tabular format for systematic comparison. Models with superior performance and independence were chosen for ensemble learning, facilitated by a stacking ensemble technique. Model parameters underwent refinement through hyperparameter tuning using random search methods. This process culminated in the generation of informative visualizations, including confusion matrices and ROC curves, to comprehensively evaluate model performance.

For text classification, the methodology employed Optical Character Recognition (OCR) and GloVe embedding for feature extraction. Classification tasks were carried out using Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM) models. Similar to image classification, the optimization of these models was conducted using the Adam Optimizer. Performance evaluation involved established metrics such as accuracy, presented systematically in tabular form. The selection of independent and high-performing models for ensemble learning was followed by further refinement through hyperparameter tuning. Visual exploration of model performance was facilitated by relevant plots and curves.

In combined image-text classification, features obtained from both image and text processing stages were concatenated. A Multilayer Perceptron (MLP) model was then applied for classification. The MLP served as a unified framework for processing and learning from the combined information extracted from images and textual data. By inputting both image and text representations into the MLP, the model optimized classification performance based on combined features. This integration facilitated more comprehensive and effective classification across various applications, ranging from multimedia content analysis to natural language understanding tasks. The MLP model underwent rigorous optimization using the Adam optimizer, and its performance was meticulously evaluated through various visualizations, including confusion matrices and ROC curves. These methodological steps aimed to select the most effective models for each task and optimize their performance to achieve accurate and reliable classification results. All models were trained using the training set before evaluation.

3. Results and Discussion

Confusion matrices, ROC curves, and precision-recall of different models were interpreted below. The confusion matrix provided a tabular summary of a classification model's performance, showing the model's predictions versus the actual class labels. It helped identify the model's accuracy and potential errors. Each cell in the matrix represented the count or proportion of instances classified accordingly. The diagonal elements (top-left to bottom-right) represented correct predictions, while off-diagonal elements represented misclassifications. The ROC curve was a graphical representation of a binary classification model's performance, plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) across various threshold values. It illustrated how well the model distinguished between positive and negative instances, with a higher Area Under the ROC Curve (AUC) indicating better performance. The value of the area closer to 1 indicated better performance. ROC curves and AUC facilitated the comparison of different models' performance and functioned in selecting the most suitable model.

3.1. Classification for Text Content

Figure 1 showed the confusion matrices and ROC curves of RNN. As shown in the figure, the AUC of RNN is 0.61.

DOI: 10.54254/2755-2721/76/20240603

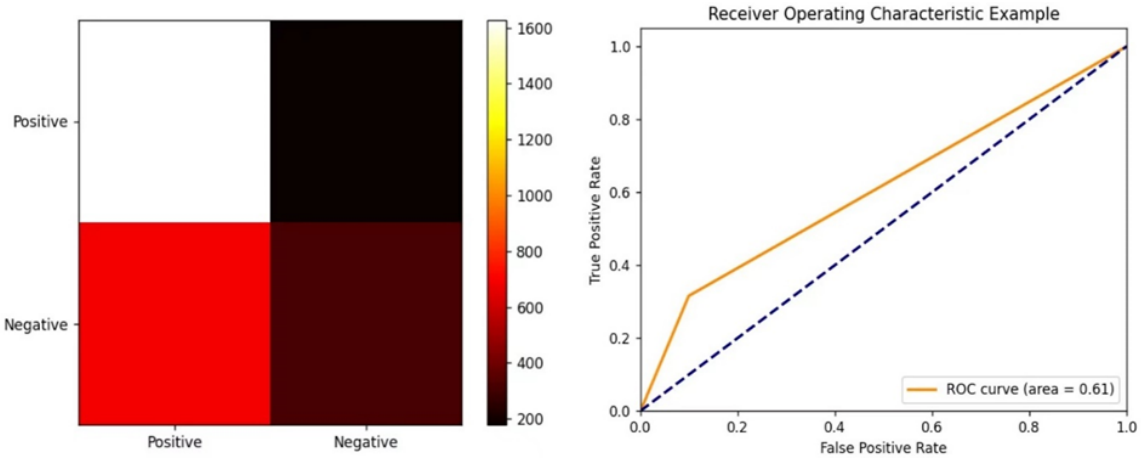

Figure 1. Confusion matrices and ROC curves of RNN.

Figure 2 showed the confusion matrices and ROC curves of GRU. As shown in the figure, the AUC of GRU is 0.60.

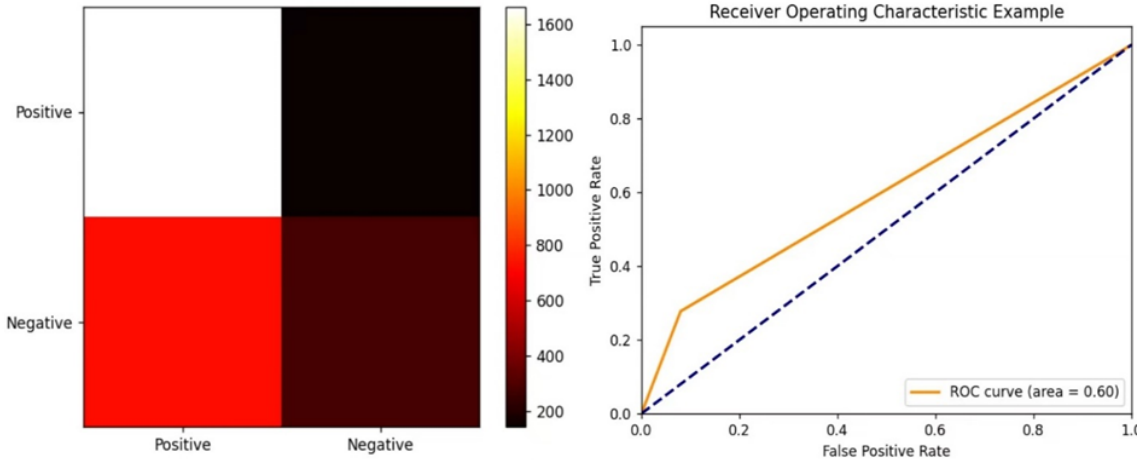

Figure 2. Confusion matrices and ROC curves of GRU.

Figure 3 showed the confusion matrices and ROC curves of LSTM. As shown in the figure, the AUC of LSTM is 0.58.

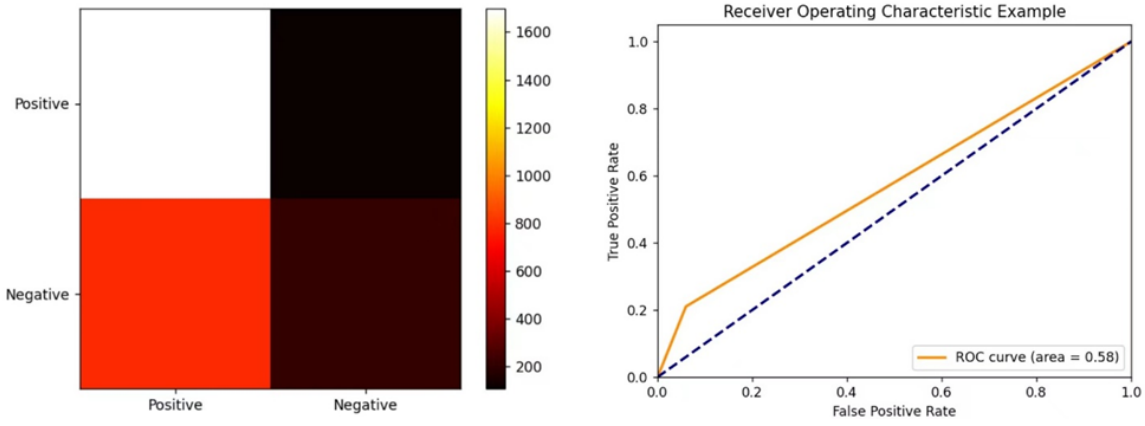

Figure 3. Confusion matrices and ROC curves of LSTM.

DOI: 10.54254/2755-2721/76/20240603

The three models, RNN, GRU, and LSTM, were evaluated. It was observed that the confusion matrices displayed similar patterns, indicating that all three models accurately predicted the negative class but exhibited some limitations in predicting the positive class. The ROC curve of LSTM exhibited the lowest elbow point, indicating its lower sensitivity compared to the other models. However, the differences were not significant.

The accuracy of the three models is summarized in Table 1. It is evident that the RNN and GRU models achieved comparable levels of accuracy, slightly outperforming the LSTM model. Nonetheless, the discrepancies in accuracy between the models were negligible, suggesting similar performance in terms of classification accuracy.

Table 1. Accuracy of RNN, GRU, LSTM models.

RNN	GRU	LSTM
0.69	0.69	0.68

Thus, it can be concluded that the performance of the three models is essentially comparable. Following evaluation, an ensemble approach was employed to combine the strengths of these models.

3.2. Classification for Image Content

Similarly, the models were measured by confusion matrix, ROC curves and accuracy. The accuracy of the three models were shown in Table 2.

Table 2. Accuracy of RF, SVM, CNN models.

RF	SVM	CNN
0.76	0.64	0.64

Table 2 shows that RF had a high level of accuracy, while Figure 4 indicates that its AUC was 0.48. AUC measures the classifier's ability to distinguish between positive and negative samples at various thresholds. Even if the classifier had an AUC of around 0.5, indicating its classification performance was equivalent to random guessing, it might still have correctly classified some samples under specific conditions, contributing to the accuracy. It is important to note that a classifier with an AUC of 0.5 could still classify some samples correctly, which could increase accuracy. However, in such cases, the classifier's predictive ability is unreliable because it might have assigned all samples to a single class, making its predictions useless.

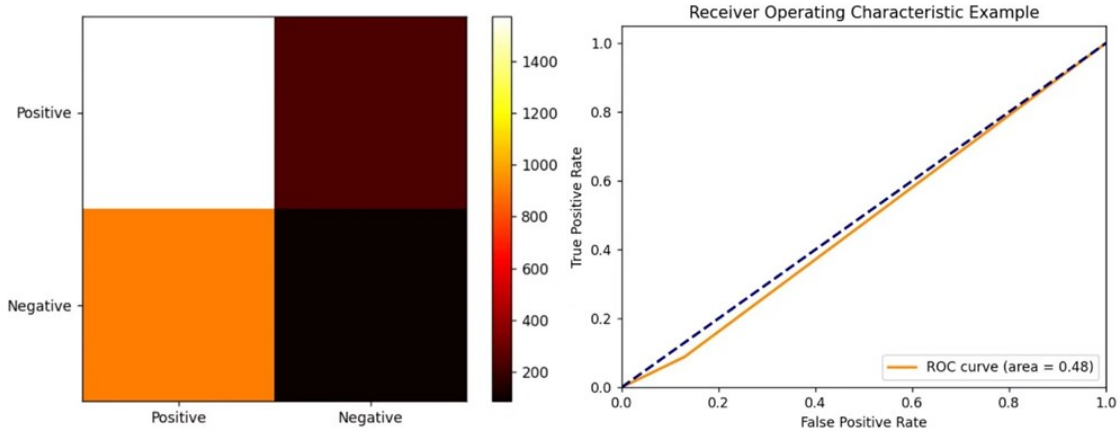

Figure 4. Confusion matrices and ROC curves of RF.

In Figure 5, the second column of the confusion matrix was entirely black, while accuracy remained at 0.64 and the AUC was 0.5. It implied the model failed to adequately discern the negative class,

DOI: 10.54254/2755-2721/76/20240603

resulting in a low accuracy rate. Moreover, an AUC of 0.5 suggests the model's predictive ability is no better than random guessing, indicating an inability to distinguish between true positive and false positive rates effectively. After evaluation, three models were ensembled.

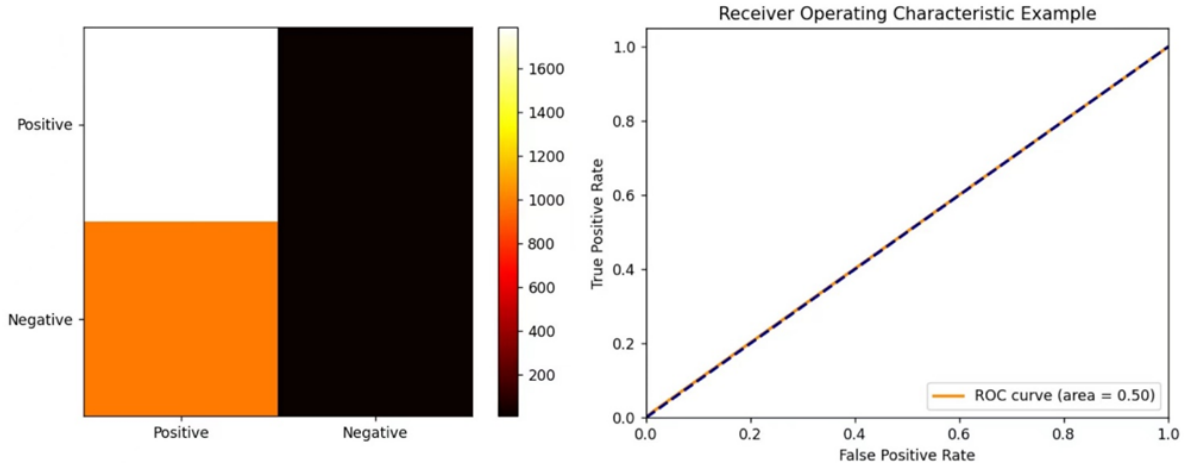

Figure 5. Confusion matrices and ROC curves of SVM and CNN.

CNN and SVM models showed same performance across various evaluation metrics, including accuracy, confusion matrices, and ROC curves. Specifically, both models achieved the same accuracy scores, indicating an equivalent level of classification accuracy. Additionally, examination of their confusion matrices revealed similar patterns of correct and incorrect classifications across different classes. The ROC curves generated for both models exhibited overlapping trajectories, indicating similar abilities to discriminate between positive and negative instances across various threshold values. These findings suggested a great similarity in the predictive performance of CNN and SVM models for the specific classification task at hand.

3.3. Combined Classification

As shown in Figure 6, compared to former results, AUC value was higher than that of image content only while lower than text content only. Compare to textual content only, combination method accurately predicted the positive class without errors, as shown by the black color in the second row's second column. However, it struggled with the negative class, as indicated by the occurrence of false positives in the second row's first column.

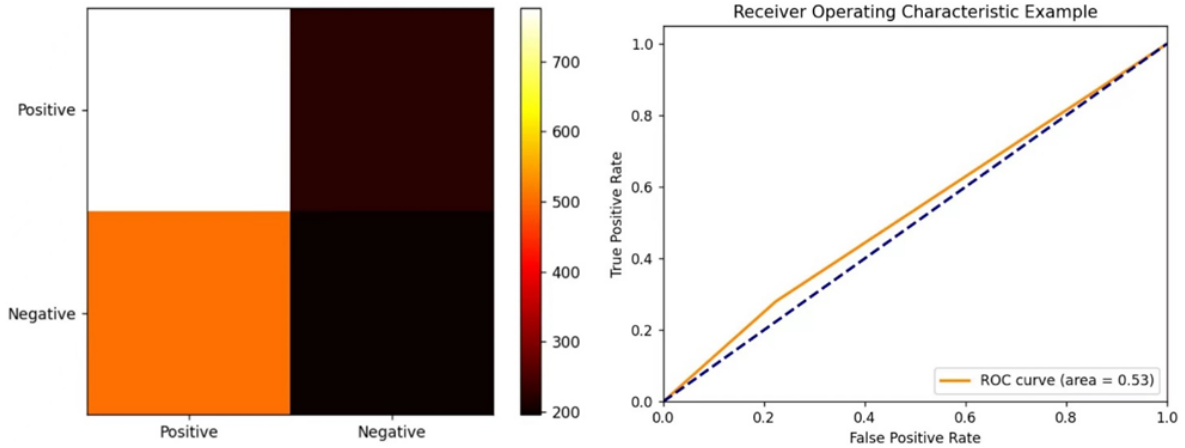

Figure 6. Confusion matrices and ROC curves of MLP

DOI: 10.54254/2755-2721/76/20240603

The accuracy of three methods was shown in Table 3. Compared to the use of image and text data separately, the combination of image and text data did not lead to a significant increase in accuracy. This observation could be attributed to several factors. First, the combination of features from different modalities has inherent challenges, which can result in redundancy or loss of information. Secondly, the quality of data and the accuracy of feature extraction are crucial. Any presence of noise or inaccuracies can adversely affect model performance. Thirdly, the selection of an appropriate ensemble technique for merging features can profoundly influence performance outcomes. Additionally, issues such as data imbalance and insufficient complementarity between image and text features may impede efforts to enhance accuracy. Therefore, achieving significant performance enhancements by integrating various modalities requires careful consideration of feature fusion strategies, ensuring data quality, and selecting ensemble methodologies judiciously.

3.4. Critical Thinking

Possible reasons for suboptimal model performance include model bias or underfitting. The model possibly lacks the capacity to capture intricate data patterns. And potential data issues include inaccuracies in labels and incomplete features. To address these problems, further investigation into the model's training process and data quality is necessary, along with potential adjustments in feature engineering or model optimization strategies.

4. Conclusion

In conclusion, the combined approach did not result in a significant performance improvement compared to individual modalities. However, it demonstrated a notable ability to identify unfriendly content. This suggests that the integration of multiple modalities has the potential to enhance detection accuracy, despite posing challenges in feature fusion and model optimization. Further investigation into refining feature representation, exploring advanced fusion techniques, and optimizing model architectures could potentially unlock greater performance gains. These findings underscore the complexity inherent in meme detection and emphasize the ongoing imperative for innovative approaches to mitigate the proliferation of harmful content on social media platforms.

References

[1] Giri, R.K., Gupta, S.C. and Gupta, U.K. (2021) An approach to detect offence in Memes using Natural Language Processing (NLP) and Deep learning. 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 1-5.

[2] Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017) Automated Hate Speech Detection and the Problem of Offensive Language. In Proceedings of the 11th International AAAI Conference on Web and Social Media.

[3] Tufekci, Z. (2018) Big questions for social media big data: representativeness, validity and other methodological pitfalls. In ICWSM.

[4] Waseem, Z. and Hovy, D. (2016) Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In Proceedings of the NAACL.

[5] Fortuna, P., Nunes, S. and Rodrigues, F. (2018) A Survey on Automatic Detection of Hate Speech in Text. ACM Computing Surveys, 51(4), 1-30.

[6] Khan, S.S., Arif, F. and van der Schaar, M. (2020) Explainable multimodal fusion by dynamic memory augmented networks for hate speech detection in social media. In Proceedings of the AAAI Conference on Artificial Intelligence.

[7] Zhang, Z., Shen, W., Zhou, M., Zhang, J., and Chen, Y. (2021) Large-scale semi-supervised representation learning for hate speech detection. In Proceedings of the AAAI Conference on Artificial Intelligence.

[8] Gehl, R. and Kluver, R. (2016) Image bites: memetic politics in the digital public sphere. New Media & Society, 18(8), 1872-1889.

DOI: 10.54254/2755-2721/76/20240603

[9] Ameer, H., et al. (2023) Multimodal Religiously Hateful Social Media Memes Classification based on Textual and Image Data. ACM Trans. Asian Low-Resour. Lang. Inf. Process.
[10] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems.
[11] Ren, S., He, K., Girshick, R. and Sun, J. (2015) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Advances in Neural Information Processing Systems.
[12] Parth, C., (2020). Facebook Hateful Meme Dataset. Retrieved from https://www.kaggle.com/datasets/parthplc/facebook-hateful-meme-dataset/data.
[13] Andrii, S., (2020). Hate Speech and Offensive Language Dataset. Retrieved from https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset.