International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 03 | Mar 2026 www.irjet.net p-ISSN: 2395-0072

Cyber Owl: A Multimodal Edge-Computing Framework for Filtering Hostile Discourse and Sensitive Imagery

ATHARVA WAGH[1], MOHD SAQLAIN[2], MOHD RAFE[3], MUSAYYAB MULLA[4], Manasi Shimpi[5]

[1]*Atharva wagh, Frontend developer (Team Leader)*
[2]*Mohd saqlain, Backend developer*
[3]*Mohd rafe, Database configurator*
[4]*Musayyab mulla, networking manager*
[5]*Manasi Shimpi Professor (Team Mentor), Dept. of computer, Abdul Razzak Kalsekar Polytechnic,Panvel, Maharashtra, India*

---***---

Abstract - *The Cyber Owl ecosystem is a privacy-centric, cross-platform parental monitoring system that is intended to monitor abusive language and explicit visual content in real-time. To address the privacy issues of cloud-based moderation, the system uses an "Intelligence Localized" architecture, where all sensitive audio and screen data is processed entirely within the Windows machine of the child. A stealthy background daemon captures system audio using loopback capture and screen frames, which are then passed through a highly optimized CPU-friendly edge machine learning pipeline, which includes Speech-to-Text, a four-stage NLP classifier, and CNN-based NudeNet. When sensitive content is detected, a concurrent Flask and SocketIO backend acts as a broker, relaying encrypted notifications to a remote Flutter-based Android dashboard. This is a comprehensive system that enables parents to receive real-time threat notifications, historical data, and remote device controls, all while maintaining robust child safety with data privacy.*

Key Words: Cyber Owl, Content Moderation, Child Safety, Privacy-Preserving, Edge Computing, Multimodal Detection, Real-Time Monitoring, Abuse Detection, Nudity Detection, Natural Language Processing , Computer Vision, Edge AI, Speech-to-Text, TF-IDF Vectorization, Support Vector Machines, Convolutional Neural Networks, Nude Net, Adhocratic Algorithm, Parental Controls, On-Device Inference

1. INTRODUCTION

The exponential growth in digital means of communication has revolutionized global connectivity but also placed vulnerable users, especially children, in serious online threats such as cyberbullying, harassment, and explicit visual media. These online threats necessitate proactive and intelligent solutions that can detect abusive language and nudity in real-time before they affect the user. This research proposes Cyber Owl, a privacy-preserving ecosystem for the moderation of abusive language and nudity in online media. The system operates entirely on consumer-grade edge hardware, providing a robust solution for digital parenting without compromising user data privacy.

1.1 Background workflow & System design

Traditional content moderation relies heavily on human reviewers or cloud-based automated systems, both of which present fundamental limitations. Human moderation is inherently reactive, struggles with the sheer volume of daily generated content, and inflicts a severe psychological toll on workers. Conversely, cloud-based AI solutions introduce significant privacy vulnerabilities and latency issues by continuously transmitting sensitive user media to external corporate servers. These challenges coupled with growing parental concerns and strict data protection regulations like GDPR and COPPA motivate the development of an edge-computing framework that performs sophisticated threat detection locally, effectively balancing child safety with absolute data privacy.

1.2 System Architecture

The Cyber Owl framework operates through a highly optimized, three-tiered architecture built for edge computing. The first tier is a stealth Windows background daemon that captures system audio via loopback and samples screen frames, routing them through a localized machine learning pipeline. This multimodal inference engine seamlessly combines a four-stage multilingual text classification cascade utilizing Adhocratic deterministic matching and TF-IDF vectorization with a CPU-optimized NudeNet convolutional neural network. When harmful content is identified, the second tier a concurrent Flask and SocketIO backend bridge brokers encrypted alerts and synchronizes databases. Finally, these alerts are pushed to the third tier, a remote Flutter-based Android dashboard, enabling immediate parental oversight without ever transmitting the raw audio or visual media files

1.3 Scope and Limitations

While the system shows clear signs of effective real-time detection capabilities, it also exhibits certain practical limitations. The text moderation system is based solely on speech-to-text transcriptions of system-captured audio streams and does not include direct keystroke logging or text input monitoring for encrypted text message applications. Additionally, in order to maintain low CPU utilization rates, the system's visual anomaly detection system is limited to a sampling rate of only 5 frames per second, which may lead to missed anomalies in high-framerate full-motion video streams. Also, the system's architectural validation and performance benchmarks are strictly limited to mid-range Windows-based operating system platforms.

2. Theoretical Foundations of Multimodal Detection

The Bag Of Words Model is a text representation method in which the text is represented as an unordered collection of words. This method loses grammar and word order but retains frequency information. However, this method is not efficient due to its high dimensionality and failure to account for any semantic relationships between words. Term Frequency-Inverse Document Frequency is an extension of the Bag Of Words Model in which the discriminative power of words in a document collection is used for weighting. When it comes to keyword matching using multiple patterns, the naive approach is to slide a window through the text. This is impractical due to its worst-case time complexity of O(n * m * k). On the other hand, the Aho-Corasick algorithm is a finite state automaton constructed from the set of patterns. This allows for matching all patterns in a single pass..

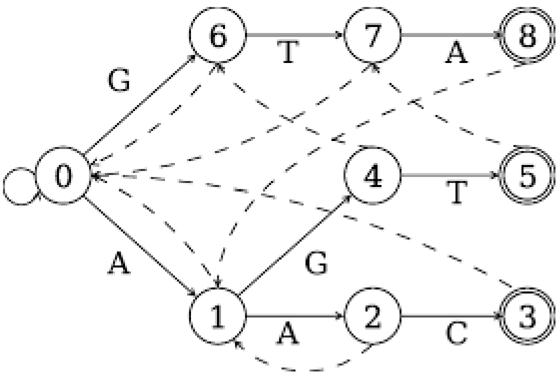

Fig -2.1: Aho-Corasick automaton structure showing the root node branching into different character states for pattern matching.

2.1 Classification Models & SVM's

Support Vector Machines (SVM) operate as supervised learning techniques that use hyperplanes to optimize class separations. During binary classification, the SVM algorithm is designed to find a hyperplane that maximizes the margin between two classes, using a regularization parameter to optimize between maximizing the margin and minimizing training error. Since SVM is not designed to provide class probabilities, Platt scaling is used to fit a sigmoid function to SVM output, providing class probabilities needed for nuanced thresholding decisions.

2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) serve as deep learning architectures specifically designed for processing grid-structured data, making them the dominant approach for visual content classification. The core mathematical operation in these networks is the convolution, wherein learnable filters slide across the input image to produce feature maps that detect specific patterns such as edges or higher-level textures.

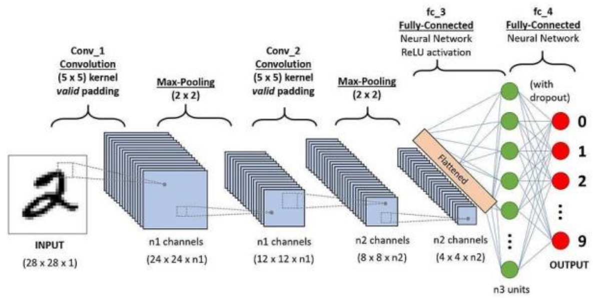

Fig -2.2: A structural diagram of a Convolutional Neural Network showing the input layer, sequential convolutional layers with filters, pooling layers,

2.3 Fusion Strategies

The multimodal fusion strategies define how to combine the data processed through separate processing pipelines. Early fusion, also known as feature-level fusion, involves concatenating features from separate modalities before a classification step, allowing cross-modal interactions to be learned but requiring intricate feature engineering to cope with heterogeneous input data. On the other hand, late fusion, also known as decision-level fusion, involves combining the outputs of separate classifiers trained on separate, modality-specific models.

2.3.1 Rules of fusion strategies

For content moderation, where sensitivity and recall are prioritized, a late fusion decision engine employing an OR rule is particularly appropriate, resulting in an unsafe classification if either the textual or visual modality exceeds a certain threshold. Visual Check: The image is of a harmless video game menu. (Score: Safe), Audio/Text Check: A severe threat is yelled in the voice chat. (Score: Unsafe), Final Decision: Safe (Visual) OR Unsafe (Text) = UNSAFE.

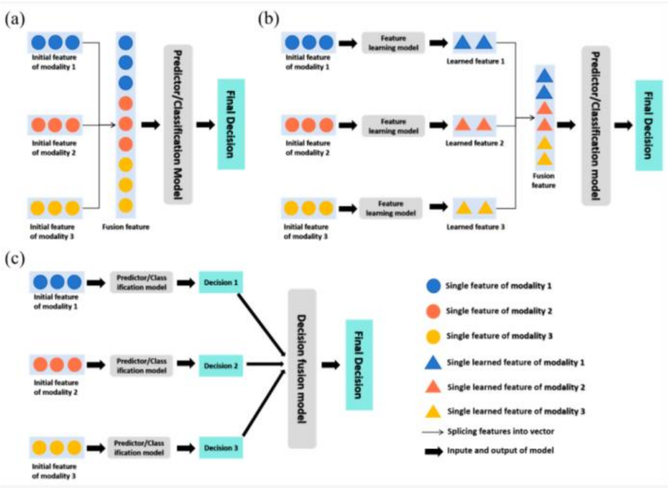

Fig -2.3: Three fusion method frameworks: (a) Early fusion framework; (b) intermediate fusion framework; and (c) late fusion framework.

3. Cyber Owl System Architecture and Ecosystem

The system design strictly follows the basic principles of modularity, concurrency, graceful degradation, and privacy by design. Modularity ensures that every module functions in isolation for easy testing and updating. Concurrency is achieved through the use of separate daemon threads for the audio and visual pipelines to take advantage of CPU utilization. A major constraint is that the analysis of any content must be performed entirely locally on the user's device without any transmission of any audio, video, text, or detection data to any external servers. Additionally, the system is subject to certain non-functional requirements that must be strictly satisfied. These include ensuring that the overall latency for the entire detection process does not exceed 500ms, CPU utilization is kept below 25%, and memory utilization does not exceed 500MB without the use of GPU acceleration

3.1 Cyber Owl System Architecture and Ecosystem

The overall architecture is a clean layered one, optimized for separation of concerns. The Data Acquisition Layer is the entry point of the system, and it is here that the system audio is captured through a loopback recording mechanism via the SoundCard library and screen captures at a rate of 5 frames per second via the MSS library.

The next layer is the Processing Layer, where the actual detection engines will be implemented, transcribing audio chunks via the Google Speech Recognition library and images via the NudeNet detection model. Finally, the overall structure of the REST API endpoint will serve as a bridge between the backend business logic and the frontend presentation layer.

3.2 Database Schema

The system utilizes a solid relational database schema that is geared towards local telemetry storage, user handling, and local caching. Some of the major entities in the schema include the USERS table for authentication, the CHILDREN table for account handling through foreign keys, and the DEVICES table for a list of registered hardware. All past detection occurrences are securely stored in the DETECTION_LOGS table, which contains details such as time stamps, source modality, label applied, and confidence level. Another table, ALERTS, manages the status of alerts. A separate KEYWORDS table is used as the basic store for multilingual abusive words for the text classification pipeline.

4. Cyber Owl System Architecture and Ecosystem

Audio Pipeline and Speech-to-Text; The first step in the textual analysis is the capture of the system's output audio using the SoundCard library. This library allows for loopback recording across platforms. The response time is reduced by segmenting the captured audio into sequential segments of 2.5 seconds before it is processed. This segmented audio is then processed using the Google Speech Recognition API, which recognizes the spoken language in real-time and provides accurate transcriptions for multiple languages, including English, Hindi, and Mandarin.

4.1. Four-Stage Text Cascade

The transcribed text is then subjected to a four-stage classification cascade of considerable sophistication. At the first stage, Aho Corasick automata are used for identifying explicit nudity keywords. This produces an immediate match with a confidence score of 0.99 for explicit keywords. At the second stage, language-specific automata are used for matching keywords related to abuse with a comprehensive lexicon containing 1,409 English keywords, 218 Hindi keywords, and 250 Chinese keywords at a confidence level of 0.95.

4.1.1. Vectorization

The third stage uses specialized pattern matching to detect concatenated tokens. This stage helps resolve common errors in speech recognition systems whereby individual words are incorrectly combined. Thereafter, text that has gone through the first three stages is vectorized and classified by a Linear Support Vector Classifier with a threshold of 0.70.

4.1.1 Stage 1: Deterministic Explicit Keyword Automata

This section will discuss the mathematical underpinnings and implementation specifics of the Aho-Corasick algorithm in relation to the first defensive layer and elucidate how high-severity explicit words such as “porn” and “nude” bypass computationally expensive machine learning models and result in a guaranteed confidence level of 0.99 in a timely manner.

4.1.2 Stage 2: Multilingual Abuse Lexicon Matching

This section describes how the Aho-Corasick pattern matching system is extended to include a large database of 1,877 words. It also describes how it handles lower-tier abusive words in English, Hindi, and Chinese by giving a confidence level of 0.95 when it finds bullying or severe profanity words.

4.1.3 Stage 3: ASR Concatenated Token Resolution

The current subsection deals with a special problem encountered by engineers. Specifically, the speech-to-text engines tend to treat rapidly uttered words as a single word (e.g., the string "shutup"). This section explains the special pattern-matching logic that has been developed for the purpose of avoiding such false negatives.

4.1.4 Stage 4: TF-IDF and LinearSVC Probabilistic Scoring

For those texts that cannot be matched by a simple keyword search, this final step serves as a catch-all mechanism. This section describes how the vectorization process using TF-IDF occurs, followed by a classification process using a Support Vector Machine and Platt Scaling to determine whether the phrase meets a 0.70 probability requirement for abuse.

4.2. Privacy Constraints and Real-Time Systems

However, there are considerable constraints involved in implementing such content moderation systems, especially in terms of latency, throughput, as well as user privacy concerns. For instance, real-time content moderation requires efficient implementation to achieve sub-second detection latency, which often requires developers to use knowledge distillation to compact large teacher models into more manageable student models. Moreover, there are considerable ethical as well as legal privacy concerns associated with transmitting user communications to cloud servers.

To mitigate the inherent privacy risks associated with transmitting potentially sensitive media to external corporate servers, contemporary parental control applications often attempt to conduct content analysis entirely on-device.

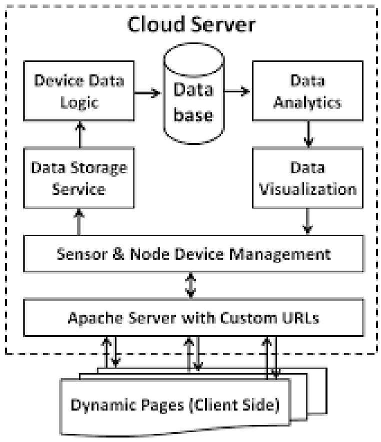

Fig -4.1: A conceptual diagram comparing a vulnerable cloud-processing architecture (data streaming to servers) with a secure on-device edge-processing architecture (data remaining entirely on the local machine).

4.3. Identified Research Gaps

While significant advancements have been made in the field of deep learning and the integration of multimodal approaches, there are several research gaps that the current literature has failed to address. One of the major issues with the current academic models is that they are based on the premise that the researcher has the luxury of using a GPU and other such computational tools. This is not the case when the focus is on the efficient design requirements for the average home computer.

4.3.1. Limitations regarding real time NLP

Moreover, there is a lack of proper focus on real-time audio-based abuse detection, and there is also a lack of support for multilingual or code-mixed linguistic varieties. Also, most commercial parental filtering tools use cloud computing for processing, which contradicts the need for privacy-preserving filtering. This gives rise to the need for an integrated local CPU-optimized multimodal framework.

4.3.2. Challenges in Low-Latency Multimodal Fusion and Contextual Nuance

Another critical gap in the current literature lies in the effective synchronization and fusion of multiple data streams under strict hardware constraints. While heavy, cloud-based models can process audio, text, and visual inputs simultaneously to understand complex interactions—such as sarcasm, rapidly evolving internet slang, or contradictory audio-visual signals—lightweight, CPU-bound models often struggle with this contextual nuance. Furthermore, accurately aligning these diverse modalities in real-time

5. Literature review

5.1 Text-Based Abuse Detection

5.1.1 Evolution from Lexicon to Machine Learning Approaches

Early forms of automated abuse detection systems used "bad words" or a "blocklist/lexicon" approach. While these were simple, precise, yet lacking, there were many problems with these approaches. First, there were many false positives, where "bad words" were found in harmless contexts, and many false negatives, where attackers would use tricks such as typos to avoid these "bad words." To solve these issues, researchers began to use machine learning techniques, such as feature engineering and Support Vector Machines (SVM) with character n-grams, making the system much harder to evade by attackers.

5.1.2 Deep Learning and Transformer-Based Models

The advent of deep learning, and more specifically sequence-based deep learning networks, saw an enormous accuracy improvement in NLP. CNNs and LSTMs eliminated the requirement for feature engineering by learning the text's deep representations directly from the data. This naturally led to the development of more accurate abuse detection transformers such as BERT. BERT's use of bidirectional pre-training resulted in unprecedented accuracy in abuse detection. However, the number of parameters in such deep learning networks is extremely high and requires significant GPU support for real-time execution.

5.2 Visual Content Moderation

5.2.1 Traditional Computer Vision and Skin-Color Heuristics

Early methods for finding explicit content relied heavily on classical computer vision techniques such as skin color models and handcrafted features that attempted to identify body parts. The theory behind it was simple: explicit content would be identifiable by its clear skin tones that could be easily identified and flagged. However, these methods were simple and light-weight but ran into hard walls.

They failed to account for the entire spectrum of human skin tones that were not well represented in these models, threw away too many innocent close-ups of faces due to misclassification as "danger," and lacked any real notion of context to identify safe skin tones vs. explicit content.

5.2.2 The Shift to CNNs and Granular Object Localization

The advent of the AlexNet breakthrough completely changed the face of the visual moderation world and marked the beginning of the meteoric rise of Convolutional Neural Networks (CNNs) for the analysis of images. The first deep learning-based classifiers, such as Yahoo's Open NSFW, provided a single probability score that reflected the overall appropriateness of an image. Today, the best approaches have moved beyond the simplistic single-label classification and into more sophisticated object detection pipelines. For instance, the NudeNet tool uses such advanced architectures to detect and locate the exact exposed anatomical features with precise bounding box coordinates and confidence levels for each feature type. This allows for extremely nuanced and flexible filtering based on the exact type of exposure rather than broad and sweeping prohibitions against entire categories of images.

5.3 Privacy Constraints and Real-Time Systems

5.3.1 Latency and Computational Bottlenecks in Live Moderation

The need to regulate the content as it's created, in real-time, puts very strict computational constraints on the design of the system. In order for the content detection to be effective in protecting the user, the detection needs to be performed in a matter of milliseconds to a few seconds after the content creation—so that the problematic content can be stopped before the user even sees it. However, these strict latency requirements put very strict constraints on the complexity of the deep learning models that can be used. In order to achieve real-time content detection without slowing down the entire system, researchers have had to resort to very aggressive optimizations, like knowledge distillation, which compresses a large, slow teacher model into a small, efficient student model.

5.3.2 Ethical Imperatives of On-Device Processing

Essentially, content moderation requires an in-depth dive into the personal conversations of users, which in itself is a major ethical and legal dilemma in the matter of privacy. The traditional cloud-based system makes this situation worse by constantly streaming user audio, text, and image data to remote corporate servers for processing.

5.3.2.1. Vulnerabilities and Compliance Risks in Cloud Transmission

This continuous transmission of sensitive multimodal data—ranging from ambient room audio to private text exchanges and screen captures—creates multiple points of vulnerability. Once data leaves the local device, it becomes susceptible to interception during transit and exposes users to potential data breaches at the remote server level. Furthermore, the indefinite storage and opaque analysis of this data by third-party entities raise significant legal compliance issues.

6. Conclusion

6.1 Constraints of Audio-Centric NLP Moderation

6.1.1 The Absence of Keystroke and Direct Text Interception

The major problem with the Cyber Owl text classification system lies in its architecture based on system audio loopback. The system's pipeline is based solely on speech-to-text recognition of the captured chunks of the system's audio. This means that anything the user types manually and silently, without the use of speech-to-text functionality, will never be picked up by the NLP system. This means that if the child is using an end-to-end encrypted messenger service, chatting in a text-based game, or reading online forums without screen readers enabled, they will never be monitored by the system. This is because the system's designers wanted it to not become an invasive form of a keylogger and thus violate the "Intelligence Localized" principle.

6.1.2 Automatic Speech Recognition (ASR) Dependencies and Errors

However, depending on audio transcription also introduces a range of vulnerabilities, depending on the level of accuracy and reliability of the speech recognizer. For example, noise, cheap mics, multiple people talking in multiplayer chat, and thick regional accents can affect the level of transcription. The system also attempts to mitigate the effects of merging errors with concatenated token resolution. However, the NLP can only perform as well as the data it is given. If the ASR engine misinterprets a word or a slurred word, then the Aho-Corasick automata and the TF-IDF models are essentially being fed bad data, which of course increases the likelihood of a missed detection in the audio channel.

6.2 Temporal Vulnerabilities in Visual Sampling

6.2.1 The 5-FPS Processing Bottleneck

To ensure that the average CPU usage remains firmly under 25%, and the memory usage doesn't exceed 500 MB without the use of a dedicated GPU, the visual detection pipeline was heavily constricted. The end result was the imposition of a fixed frame rate of precisely 5 frames per second. While the current state of digital video and games maintains a minimum frame rate of 30-60 FPS, the Cyber Owl system is only concerned with a small percentage of the screen's visual data. There is a clear temporal gap between the time an image is displayed on the screen and the time the NudeNet CNN is capable of detecting it: "A very explicit or harmful image, such as a quickly scrolled social media feed or a video frame, could appear on the screen for a moment before passing through the 200ms capture windows without being detected by the NudeNet CNN."

6.2.2 Contextual Blindness in Isolated Frame Analysis

The visual pipeline works on the downsampled JPEG images of size 640*480 individually, without referencing the images before or after the current one. It doesn't benefit from the context of the images in terms of time. The NudeNet model identifies the exposed anatomical features in the grid of images, but it doesn't process the images in a sequence of frames to comprehend the content in a larger sense. :

6.2.2.1. Limitations of Static Feature Extraction

By intentionally discarding recurrent architectures and 3D convolutions to maintain CPU efficiency, the visual pipeline evaluates each frame as an isolated event. Consequently, the trajectory, pacing, and evolving narrative of a video stream are entirely lost. While the system reliably detects features in a frozen moment, it remains completely detached from the sequential events that define the media's true intent.

6.2.2.2. Semantic Ambiguity and False Positives

This lack of temporal continuity creates significant semantic ambiguity, resulting in high false-positive rates for benign material. The context-free visual engine assigns the same threat level to a clinical biology documentary as it would to explicit pornography, as both generate identical bounding box signals at the frame level. Without evaluating sequential frames, the edge-optimized system struggles to differentiate educational context from harmful content.

7. Limitation of review

This research, while demonstrating robust edge-computing capabilities for privacy-preserving content moderation, is bound by several inherent architectural and operational limitations that prioritize real-time efficiency over exhaustive data capture. Primarily, the natural language processing pipeline is strictly audio-centric, relying entirely on speech-to-text transcriptions captured via system audio loopback; this renders the framework functionally blind to direct keystrokes and text-based communications within end-to-end encrypted messaging applications, while also introducing vulnerabilities related to automatic speech recognition (ASR) transcription errors caused by background noise or heavy accents. Furthermore, to adhere to strict CPU and memory constraints without requiring dedicated GPU acceleration, the visual detection pipeline is aggressively locked to a 5-frames-per-second sampling rate.

This temporal bottleneck, combined with the Convolutional Neural Network's evaluation of isolated static frames without recurrent sequential context, creates blind spots where split-second explicit anomalies in high-framerate multimedia may evade detection. Additionally, the system's current programmatic implementation is heavily coupled to the Windows operating system and demands a baseline of mid-range computational power (such as an Intel Core i7 with 16GB of RAM), effectively precluding native deployment on macOS, Linux, or highly resource-constrained microcomputers without inducing severe thermal throttling or interface lag.

8. References

1) Davidson, T., Waring, W., Pomarole, U., & Mejova, Y. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. Proceedings of the International AAAI Conference on Web and Social Media.

2) Waseem, Z., & Hovy, D. (2016). Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection. Proceedings of the NAACL Student Research Workshop.

3) Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017). Deep Learning for Hate Speech Detection in Tweets. Proceedings of the 26th International Conference on World Wide Web.

4) Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

5) Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., & Chang, Y. (2016). Abusive Language Detection in Online User Content. Proceedings of the 25th International Conference on World Wide Web.

6) Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems.

7) Jones, M. J., & Rehg, J. M. (2002). Statistical Color Models with Application to Skin Detection. International Journal of Computer Vision.

8) Deselaers, T., Gass, T., Dreuw, P., & Ney, H. (2008). Bag-of-Visual-Words Models for Adult Image Classification and Filtering. 2008 19th International Conference on Pattern Recognition.

9) Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

10) Gomez, R., Gibert, J., Gomez, L., & Karatzas, D. (2020). Exploring Hate Speech Detection in Multimodal Publications. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.

11) Aho, A. V., & Corasick, M. J. (1975). Efficient String Matching: An Aid to Bibliographic Search. Communications of the ACM.

12) Salton, G., & Buckley, C. (1988). Term-Weighting Approaches in Automatic Text Retrieval. Information Processing & Management.

13) Joachims, T. (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. European Conference on Machine Learning.

14) Platt, J. (1999). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. Advances in Large Margin Classifiers.

15) Kiela, D., Firooz, H., Mohan, A., Goswami, V., Singh, A., ... & Joulin, A. (2020). The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes. Advances in Neural Information Processing Systems.

16) Kumar, S., Spezzano, F., & Subrahmanian, V. S. (2019). Real-time Hate Speech Detection System for Live Streaming. IEEE International Conference on Data Mining.

17) Han, S., Mao, H., & Dally, W. J. (2015). Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. International Conference on Learning Representations.

18) Lane, N. D., Bhattacharya, S., Georgiev, P., Forlivesi, C., Jiao, L., ... & Kawsar, F. (2016). DeepX: A Software Accelerator for Low-Power

19) O'Keeffe, G. S., & Clarke-Pearson, K. (2011). The Impact of Social Media on Children, Adolescents, and Families. Pediatrics.