

Deep Learning Driven Web Security: Detecting and Preventing Explicit Content

Ganeshayya Shidaganti, Shubeeksh Kumaran, Vishwachetan D, Tejas B N Shetty

Dept. of Computer Science and Engineering, M S Ramaiah Institute of Technology, Bangalore, Karnataka, India

Abstract—In today's digital age, the vast expanse of online content has made it increasingly accessible for users to encounter inappropriate text, images and videos. The repercussions of such exposure are concerning, impacting individuals and society adversely. Exposure to violent content can lead to undesirable human emotions, including desensitization, aggression, and other harmful effects. We utilize a machine learning approach aimed at real-time violence detection in text, images and videos embedded in the website. The foundation of this approach lies in a deep learning model, highly trained on a vast dataset of manually labeled images categorized as violent or non-violent. The model boasts exceptional accuracy in identifying violence in images, subsequently filter out violent content from online platforms. By performing all processing intensive tasks in the Cloud, and storing the data in a database, an improved user experience is achieved by completing all the necessary detection processes at a lower time frame, and also reducing the processing load on the user's local system. The detection of the violent videos is done by a CNN model, which was trained on violent and non-violent video data, and the detection of emotions in the text is taken in by a NLP based algorithm. By implementing this highly efficient approach, web safety can undergo a significant improvement. Users can now navigate the web with confidence, free from concerns about accidentally encountering violent content, fostering improved mental health, and cultivating a more positive online environment. We are able to achieve 67% accuracy in detecting violent content at approximately 2.5 seconds at its best scenario.

Keywords—Web safety; machine learning; cloud computing; natural language processing; web-scraping, big data

I. INTRODUCTION

The Internet has become an integral part of modern life, transforming the way we access information, communicate, and interact with the world. However, with its many advantages, the digital realm also presents various challenges, especially in terms of ensuring network security for users. One of the greatest concerns is the proliferation of overt cases of violence which pose a significant risk, particularly to vulnerable individuals such as children and adolescents. To address this critical issue, we propose a novel approach that uses machine learning techniques, specifically a natural language processing technique for text processing and deep learning Convolutional Neural Network (CNN), propose real-time disturbance detection in images and videos. The unrestricted access to the internet has led to an increase in explicit violent content on online platforms. This disturbing trend poses serious risks to users, ranging from psychological injury to desensitization to violence. In addition, children and young people exposed to such substances that can have long-

term negative effects on their mental and emotional well-being. Existing manual filtering methods and traditional keyword-based methods are often unable to effectively identify and prevent violent events, requiring advanced automated solutions. Insights into how the Internet has changed as a communication tool and the expanding demand for site filtering services was inferred from [1]. The suggested real-time violence detection system, which aims to combat dangerous and inappropriate information on the Internet, was motivated by this concept. From [2] a deep understanding of the difficulties brought about by the enormous volume of data and the persistent infractions on social media platforms can be obtained. The decision to use deep learning for real-time violence detection was motivated by this realization because it would effectively filter content without only depending on human interaction. [3] Shed some light on the growing use of algorithmic content filtering systems for popular websites like Facebook, YouTube, and Twitter. In response to the growing demand for efficient content filtering, we decided to use similar technical methods for real-time violence detection based on this knowledge.

The inspiration from [4] which suggested classifying web pages using artificial neural networks as a part of content filtering. In the beginning we started with a deep learning Convolutional Neural Network (CNN) model for identifying explicit information in web photos as we realized the potential of neural networks for in-the-moment content analysis. Comparative scarcity of research on detecting aggressive/violent behaviors and fights within video content was highlighted in [4]. This knowledge informed the choice to concentrate on violence detection since we recognized its usefulness in situations like the proposed real-time web safety system, when identifying explicit content is crucial. Inspired by the proposed method's utilization of extreme acceleration patterns as discriminant features in [5], we used similar ideas into the CNN model. By implementing this method, the accuracy of the violence identification by a large margin and displayed the potential of using the unique qualities of violent behaviors for better content filtering. The goal of developing a highly accurate and efficient violence detection system was in line with [6]'s emphasis on the use of local spatio-temporal elements in characterizing multimedia information. We were able to improve the system's capacity to differentiate between violent behaviors and regular activities by incorporating this understanding the CNN model, hence enhancing content filtering precision.

The main objective of this research is to enhance web security using a real-time violence detection system that

automatically filters and filters images with obvious violent content using the power of machine learning, we aim to create efficient and accurate solutions. The proposed method has two main steps. In the first step, a web page listener on the user's machine picks up the link from the visited web page and sends it to the server. The server checks its database to see if the website has been previously tagged or indexed for explicit content. If the content is available, it is immediately redirected to the user's device, and if there is violent content, the site is invisible. In the second stage, information about web pages that are not listed in the database is forwarded to the cloud computing service. There, images from various websites are processed using CNN's deep image learning algorithm, which is trained to detect clutter in images. The search results are then sent back to the server, where the web page information is added to the database. The results are then transmitted to the user's device, making clear images unrecognizable in real time. The proposed machine learning-based violence detection system has a great potential to significantly affect web security and protect users from obviously harmful content through the violence detection method an automated and incorporating real-time responses, the system ensures instant protection and enables users to have a secure online experience. Furthermore, the research contributes to the growing field of applying machine learning techniques to content filtering, paving the way for more advanced solutions in the future.

II. LITERATURE REVIEW

Through a thorough study of important research papers, this literature review aims to highlight the symbiotic link between deep learning, cloud computing, and real-time communication. Collectively, the chosen papers highlight how these technologies have a revolutionary effect on a variety of applications, shedding light on how their integration promotes creativity, improves efficiency, and lessens difficulties in a variety of fields.

The methodology was heavily motivated by the [7]'s classification of violence detection algorithms based on conventional machine learning, support vector machine (SVM), and deep learning techniques. Within the limits of the framework offered by the evaluated classification techniques, we customized and updated the deep learning CNN model for violence detection. The research in [8] had a unique approach of eliminating violent scenes from movies through a three-step process that stood-out with the content filtering objective. To ensure quick and accurate detection of explicit content, we utilized a similar technique of analyzing frames and applying deep learning for violent detection, but in the context of web pages as opposed to movies.

The method of using a deep learning CNN model for violence detection was motivated with the research involving fully connected networks and long short-term memory (LSTM) networks for frame-level violence detection as seen in [9]. The idea of showing the features through attention mechanisms influenced the enhancement approach for recognizing explicit/violent content in real-time, even though the focus is on web content and not video frames. The development of models for accurate face recognition, such as the multi-foot input CNN model and the SPP-based CNN model in [10],

combined with the view on picture analysis and identification in the real-time violence detection system. The idea of improving accuracy through specialized CNN models influenced the optimization methods for content filtering, even though the concentration is on content rather than faces. The urge to develop a system that reduces such mistakes was influenced by [11]'s acknowledgment of the difficulty presented by false positives in violence detection, evidently those brought on by friendly behaviors. The strategy for improving the accuracy and applicability of the real-time system was inspired by the paper's presentation of three deep learning-based models for violence detection, including those based on transfer learning and training from scratch. The optimization methods for developing a highly reliable and effective violence detection system were influenced by the [12]'s assessment of improvements in CNNs, including layer design, activation functions, loss functions, regularization, optimization, and fast computation.

The understanding of the difficulties involved in delivering an easy-to-use access to information in different situations was influenced by [13] in examination of the mobility features in mobile client-server computing. This influenced how we approached developing a real-time violence detection system that takes into consideration the various situations in life in which people access web material. The study in [14] highlighted the critical working of client-server systems in information technology view, including a range of tasks like web-based applications, unified computing, mobile apps, and cloud computing. We understood the importance of both logical and physical components, such as programming scripts and networking, in providing effective and accurate content filtering. The architectural design choices were influenced by the [15]'s study of issues including the in-depth learnings of responsibilities between clients and servers and the thinking of client/server systems. In order to ensure the effectiveness and efficiency of the real-time violence detection system, we understood how important it was to assign defined tasks and organize the system to become more efficient. The overarching objective revolves around ensuring a seamless and efficient link between the real-time violence detection system and the clientele. This objective harmonizes with the insights from study [16], which illuminates the role of Web sockets as a conduit for establishing dynamic two-way communication within HTML5 compliant browsers. The emphasis on the singular socket prowess of Web sockets profoundly informed the system's design blueprint. Notably, the techniques for optimizing the construction of a real-time violence detection system drew inspiration from the principles elucidated by the works of [17]. These principles, underscored by a devotion to minimal latency and expansive scalability in the realm of web sockets, were intricately woven into the developmental fabric.

Light is shed on cloud computing's transformational effects and its ability to use machine learning methods in [18]. This understanding is in line with how we use cloud computing to combine deep learning CNN models for real-time violence detection and content filtering. The idea of using cloud computing for deploying and managing deep learning models was in line with the [19]'s analysis of the effects of cloud computing on the area of machine learning. Its examination of

parallelization using cutting-edge parallel computing frameworks like MapReduce, CUDA, and Dryad helped us better grasp how to analyze large data sets quickly. As managing vast amount of data is essential for precise content analysis, this realization was in line with the intention to apply deep learning CNN models for real-time violence detection. The research in [20] stressed the significance of reliability in cloud computing systems, admitting that errors are unavoidable even with architectures built for high service availability. We understood the importance of locating critical characteristics that are associated with application failures because doing so would help us better manage computational resources and mitigate wasteful resource consumption as a result of failed tasks. The study in [21] focuses on the difficulties associated with computationally effective data-driven prognostics and health management (PHM) was in line with the goal of effectively processing and analyzing massive amounts of web content in real time using cloud-based data management solutions. The research in [22] provided a comprehensive survey, which highlighted the challenges and future directions in this area, and [23] proposed a text classification approach, achieving an excellent performance on a benchmark dataset. Table I represents the comparative analysis and limitations of existing approaches as in [24], [25], [26] that this study is trying to overcome.

TABLE I. COMPARISON AND ANALYSIS OF MODULAR APPROACHES

| Feature | [24] | [25] | [26] |
|-------------|--|--|--|
| Focus | Explicit content detection | Audio-visual-textual cyberbullying detection | Cyberbullying detection and prevention |
| Methods | Rule-based, machine learning | Deep learning | Data mining, psychological perspective |
| Findings | Rule-based methods are effective for detecting explicit content, but machine learning methods can achieve higher accuracy. | Deep learning methods can be effective for detecting cyberbullying across multiple modalities. | Data mining and psychological methods can be used to identify cyberbullying behaviour. |
| Limitations | Rule-based methods can be difficult to maintain as new forms of explicit content emerge. | Although powerful, deep learning methods can be computationally expensive. | Data mining methods can be difficult to interpret. |

The knowledge gained from these foundational publications guides the future work because deep learning, cloud computing, and real-time communication are all interconnected in this context. The lessons learned from the studied works give us a wealth of strategic viewpoints, practical suggestions and methodological strategies.

III. SYSTEM ARCHITECTURE

The system architecture is explained in three parts, Client-Side Setup, Server hosted Database, Cloud Computing as visually represented in Fig. 1.

A. Client-Side Setup

A node-JS based setup is chosen due to its ability to run on the user’s inbuilt Chrome Browser with event listeners for extraction of Text, Images and Videos as and when the data is being loaded on the application. This is achieved using the Puppeteer modules that gives us the access to the requires data even before it appears on the user’s screen. By using this module, we have more control over the happenings of this browser instance in real-time. This is a more convenient approach as opposed to other implementations where an iframe where the content of a website is being broadcasted into a container of the service provider’s website and then host the processed website in a third-party browser. This does not just make the entire process tedious, but also causes delays throughout the user’s browsing experience.

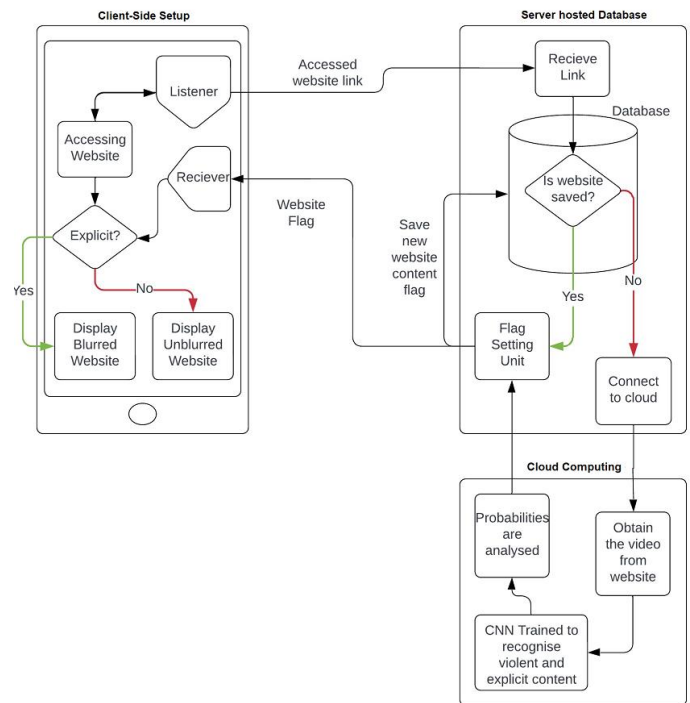


Fig. 1. System architecture.

B. Server Hosted Database

The website data in the database is accessed via a hosted server connected to the Node-JS program firing the client’s website. This server is responsible for handling URL inputs from the client and responding with safety-related information about the accessed websites either by going through the Database list, or by performing ML based Data analysis. This analysis is made on a Cloud Computer equipped to perform resource-intensive computations efficiently which directly updates the database in the Server. This dynamic process ensures that the database remains in a state of continuous enhancement, after every process and increase the accuracy over time.

C. Cloud Computing

Amazon AWS machine learning services provides both a cost effective and high-performance infrastructure. By deploying a Natural Language Processing and CNN model-

based ML programs in this platform, we achieve optimal performance to obtain a seamless user experience without utilizing local processing resources at all. AWS's Auto scaling capabilities allows the system to dynamically adjust resources based on demand, maintaining efficiency while avoiding unnecessary costs. The Amazon S3 storage solution ensures highly reliable data storage. Meanwhile AWS Lambda is a server less computing service that supports event driven executions. Key specifications and features to consider when deploying ML programs based on using Amazon AWS machine learning services are in Table II.

TABLE II. AWS FEATURES OPTED

| Specification/Feature | Description |
|-----------------------|---|
| Instance Types | Choose compute-optimized instances (e.g., C5, M5) for efficient ML processing. |
| Auto Scaling | Implement dynamic resource adjustments to match varying workload requirements. |
| Storage Solutions | Leverage Amazon S3 for reliable and scalable data storage. |
| AWS Lambda | Utilize serverless computing for event-driven ML executions. |
| Amazon Sage Maker | Explore a managed service for end-to-end ML development and deployment. |
| Elastic Inference | Optimize inference performance and cost with GPU acceleration. |
| GPU Instances | For CNN-based models, consider GPU-equipped instances for enhanced performance. |
| AWS Lambda Edge | Use edge computing for low-latency execution near users. |
| AWS Step Functions | Orchestrate complex workflows for ML data processing and analysis. |
| AWS Data Analytics | Explore data analysis services for extracting insights from ML results. |
| AWS Cost Management | Monitor and control costs with AWS Cost Explorer and Budgets. |

IV. METHODOLOGY

In this section, we outline the step-by-step methodology employed to realize the proposed real-time violence detection system. As represented in Fig. 2, the methodology encompasses two primary phases: the web page processing phase, which determines whether a website is flagged or indexed, and the image analysis phase, where a deep learning Convolutional Neural Network (CNN) model is utilized for detecting violent content in images.

A. Web Page Processing Phase

In this initial phase, the system seeks to determine the nature of the accessed website by checking its status in the database and providing real-time feedback to the user.

1) *Web page listener and data transmission:* We implement a lightweight web page listener utilizing a Puppeteer based Node.JS approach on the user's device. This allows us to access the data flowing into the device when a web page is being requested from the client side. This enables us to grab the text, images and videos embedded in the website. When a user accesses a web page, the listener captures the webpage link and transmits it to the server through an encrypted HTTPS connection. This ensures the secure transfer of user data while maintaining their privacy

and at the same time transferring the minimal amount of information required for further processing.

2) *Database check for flags or indexing:* The server hosts a relational database containing website information, including URLs, content flags, and indexing status. When the server receives the webpage link, it queries the database to determine if the website has been flagged for violent content or previously indexed in the past processing of the system. If such information is found to be previously flagged, the relevant data is relayed back to the user's device and appropriate actions are taken on the website view in the client application immediately. This approach's motive is to bypass the necessity to perform unnecessary processing of text, images and videos for every request and hence reducing the overall processing time, therefore trying to provide a seamless user experience.

3) *Web page blur/unblur:* If the database query confirms the presence of explicit content on the website, the user's device receives a notification instructing it to blur the explicit text, images, videos using the puppeteer module's capability. This user-friendly approach minimizes direct exposure to harmful content and maintains a safe browsing environment with quick processing.

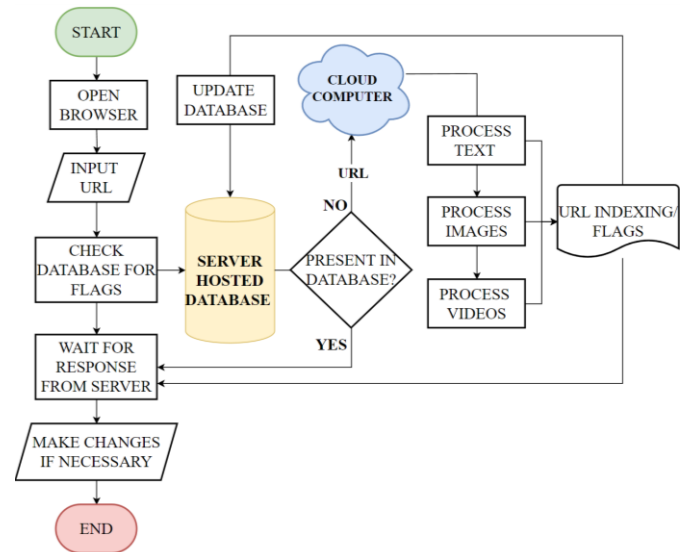


Fig. 2. Flow diagram of the system.

B. Data Analysis Phase

For websites without prior database information, the system initiates the data analysis phase, involving the application of a Natural Language Processing and deep learning CNN model to detect violent content in text and images/videos respectively.

1) *Cloud computing service setup:* To accommodate the computational demands of image analysis, we set up a cloud-based environment using Amazon Web Services (AWS). The cloud service is equipped with sufficient computational resources to handle multiple requests simultaneously, ensuring real-time responses. The major benefit of utilizing a cloud-based approach is the reduced load on the local user's system.

Hence the entire process will not affect the user’s browsing experience, additionally the other background tasks running in the client-side systems will not be pulled out of its resource allocations for this purpose.

2) *Text processing*: The text is first extracted and analyzed using NLTK modules for sentiment analysis. The Sentiment Intensity Analyzer class of the NLTK’s ‘sentiment module’ is utilized to score the text as polarity_scores. The score of all sentences is combined to form a compound score which is then classified as positive, neutral, and negative. Each sentence's polarity score is combined to generate a compound score, computed as in Eq. (1).

$$\text{Compound Score} = \sum \text{Polarity Score of Sentences} \quad (1)$$

This is done by using threshold values of -0.05 and +0.05 to categorize them as disturbing content or not as seen in Table III. Once it is determined as inappropriate, it halts the Data Analysis Phase and blurs the entire page.

TABLE III. TEXT CLASSIFICATION

| Compound Score Range | Sentiment Classification |
|----------------------|--------------------------|
| Less than -0.05 | Negative Sentiment |
| -0.05 to 0.05 | Neutral Sentiment |
| Greater than 0.05 | Positive Sentiment |

3) *Data preprocessing*: Images extracted from the accessed website's content, including frames from videos, are pre-processed to standardize dimensions and enhance the model's performance. The images are resized to 224x224 pixels irrespective to their original size or resolution or aspect ratio using Eq. (2).

$$\text{Resized Image} = \text{Resize}(\text{Original}, \text{Target}) \quad (2)$$

After resizing, the images are normalized to the [0, 1] range. This normalization process involves scaling the pixel values to fit within the specified range, which is vital for consistent input across the entire process.

$$\text{Normalized Image} = \frac{\text{Resized Image}}{255} \quad (3)$$

Eq. (3), where 255 represents the maximum pixel value in an 8-bit image. This allows that all inputs are maintained with consistency and is standard across the entire process.

4) *Deep learning CNN model*: For image analysis, we employ a Convoluted Neural Network model to analyze the images. To be specific, a pre-trained VGG16 CNN architecture, fine-tuned to detect violent content. The VGG16 model has demonstrated effectiveness in image classification tasks due to its deep 16-layer intensive ability to extract vital and meaningful information regarding the input images. By utilizing the VGG16 model to analyze individual frames extracted from videos, the system gains the ability to examine the video content frame by frame. This enables the detection of violent elements or patterns within the vide.

5) *Violent content detection*: Pre-processed images are fed through the CNN model to obtain predictions. The model outputs a probability score indicating the likelihood of the image containing violent content.

The program that analyzes the video consists of the modules and packages as seen in Table IV. Additionally, the parameters and variables assigned certain values to ensure the perfect working of this system is included in Table V.

TABLE IV. MODULES AND PACKAGES

| Modules and Packages | Description |
|----------------------|--|
| Asyncio | Employed for asynchronous programming, allowing efficient handling of WebSocket connections. |
| Websockets | Utilized to facilitate communication through WebSocket connections, crucial for real-time interaction. |
| OS | Enables essential file operations, such as handling paths and file removal. |
| PIL | Facilitates image processing tasks, aiding in resizing and other operations. |
| cv2 | Integral in reading and manipulating video frames, a key step in content analysis. |
| NumPy | Supports numerical operations, aiding in array manipulations and calculations. |

TABLE V. PARAMETERS AND VARIABLES

| Parameters and Variables | Description |
|------------------------------|---|
| CLASSES_LIST | An array of class labels, specifically "Non-violence" and "Violence," used for classification. |
| model | The pre-trained CNN model loaded using Keras' load_model function, essential for content analysis. |
| SEQUENCE_LENGTH | A parameter determining the number of frames utilized for sequence analysis within the CNN model. |
| IMAGE_HEIGHT and IMAGE_WIDTH | Parameters dictating the dimensions to which frames are resized, ensuring consistency during analysis. |

To classify an image as violent or not, we set a threshold of 0.5 as in Table VI, based on validation results, above which an image is categorized as violent. The CNN model assigns a probability score $F_{\text{violent}P_{\text{violent}}}$ to each image. This score represents the likelihood of the image containing violent content. It can be expressed as in Eq. (4).

$$F_{\text{violent}P_{\text{violent}}} = \text{CNN}_{\text{Model}}(\text{Pre-processed Image}) \quad (4)$$

TABLE VI. VIDEO CLASSIFICATION

| Probability Score | Classification |
|-------------------------------|---------------------|
| $P_{\text{violent}} > 0.5$ | Violent Content |
| $P_{\text{violent}} \leq 0.5$ | Non-Violent Content |

6) *Result transmission and database update*: The predictions from the CNN model are transmitted back to the server, which subsequently updates the database with the

violence detection result for the accessed website. This dynamic database enhancement ensures the continual improvement of the system's accuracy over time. This shows that the cloud is only communicating with the database and not with the client device directly. This method ensures that this process is not susceptible to external cyber-attack.

7) *User feedback and blurring*: The user's device receives the result of the violence detection process. If the result confirms the presence of explicit content, the user's device applies blurring to the corresponding images or videos, ensuring a safer browsing experience. In this case only the and all the non-violent images/videos stays unblurred, inferring that its just the specific violent content that are blurred. This is to provide the user with the best possible experience.

V. RESULTS

To demonstrate the working of this system, local websites are utilized. This system is implemented by using a local cloud instance to run the Machine Learning programs that analyzes the text, images and videos. Additionally, a MongoDB database is fired-up to store the website information. All communications are made using web sockets to best emulate the real implementational results. In this implementation, the entire content is first blurred and eventually the content safe to be viewed are unblurred ensuring that no disturbing content is viewed in the small timeframe of computation. Multiple cases are considered to portray the working of the system. The result images depict the results in case of both when it's present and not present in the database.

A. CASE 1: Non-Violent Text & No Videos in the Website

The text is classified as neutral and hence the website is unblurred as seen in Fig. 3.

| WEBSITE (NOT BLURRED) | |
|--|--|
| Example Domain | |
| This domain is for use in illustrative examples in documents. You may use this domain in literature without prior coordination or asking for permission. | |
| More information... | |
| PROGRAM LOGS | |
| NOT-PRESENT IN DATABASE | PRESENT IN DATABASE |
| page blurred Downloaded text Python script output: neutral Unblurred No videos | page blurred Downloaded text Present in database WebSocket client connected. <Buffer 75 6e 62 6c 75 72> Unblurred |

Fig. 3. Non-violent text and no videos in the website.

B. CASE 2: Violent Text & (No/Violent/Non-Violent) Videos

The text is classified as negative and no other content is considered (videos). Hence the website is completely blurred as in Fig. 4.

C. CASE 3: Non-Violent Text & Non-Violent Videos

The text is classified as neutral, the video is declared positive. Hence the website is unblurred as in Fig. 5.


| WEBSITE (BLURRED) | |
|--|--|
|  | |
| PROGRAM LOGS | |
| NOT-PRESENT IN DATABASE AND PRESENT IN DATABASE | |
| page blurred Downloaded text Python script output: negative | |

Fig. 4. Violent text and (no/violent/non-violent) videos.


| WEBSITE (NOT BLURRED) | |
|---|--|
|  | |
| PROGRAM LOGS | |
| NOT-PRESENT IN DATABASE | |
| page blurred Downloaded text Python script output: neutral Unblurred Videos Exist WebSocket client connected. Python script output: Received message from server: D:/Violence/sites/NV-NV.html D:/Violence/tej/NV_1005.mp4 | |
| PRESENT IN DATABASE | |
| <Buffer 75 6e 62 6c 75 72> Received unblur signal from Python client.unblur | |

Fig. 5. Non-violent text and non-violent videos.

D. CASE 4: Non-Violent Text & Violent Videos

The text is classified as neutral and video as negative. Hence the video portion of the website is only blurred as in Fig. 6.

The time taken to perform the various cases discussed is shown in Table VII. The data is visualized in the form of line graph. It can be easily noted that the Computation time is proportional to the length of the text or video being analyzed in the website. This noticed that, with the support of the database, we have continuous Time frame of two to three seconds. Except the case when the text in small and no video is present in the website, all other iterations require a considerably longer duration to process the content. Fig. 7 shows the ability to identify violent images quickly.

Even though the Fig. 8 shows 100% successful results in case of those four iterations, the model has an accuracy of 67% overall. Hence, utilizing this architecture, we are able to achieve exceptional speed and minimal computational load. Therefore, ensuring that the browser users are protected from

disturbing content, and without much interruption due to quick the shift between unblur and blur during the processing phase.

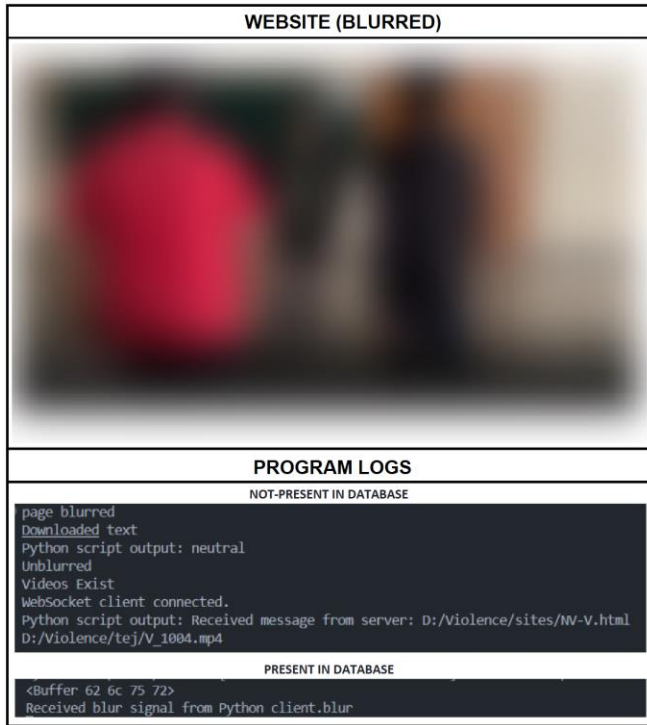


Fig. 6. Non-violent text and violent videos.

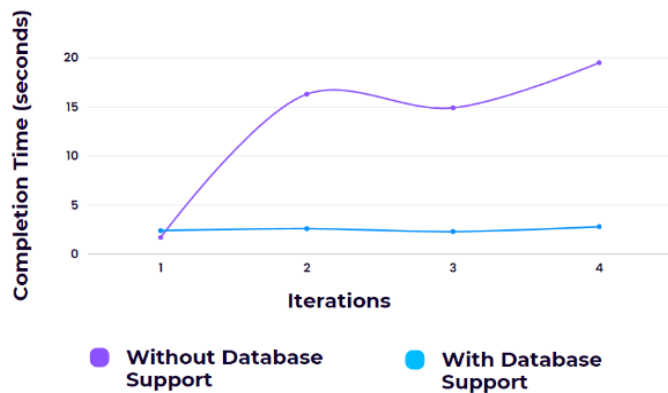


Fig. 7. Comparing the implementation time with and without database.

TABLE VII. PROCESSING TIME

| CASE | Text Processing Time | Video Completion Time | Total Time when not in database | Total time when URL is in the database | Probability Score |
|------|----------------------|-----------------------|---------------------------------|--|-------------------|
| 1 | 1.7 sec | NULL | 1.7 sec | 2.4 sec | 0.2 |
| 2 | 1.3 sec | NULL/ 0 – 15 sec | 1.3 – 16.3 sec | 2.3 sec | 0.7 |
| 3 | 1.2 sec | 13.7 sec | 14.9 sec | 2.6 sec | 0.3 |
| 4 | 1.4 sec | 18.1 sec | 19.5 sec | 2.8 sec | 0.8 |

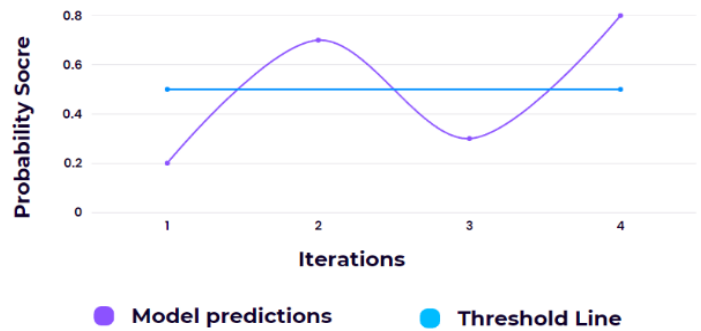


Fig. 8. Detection performance comparison graph.

VI. CONCLUSION

Considering the evolution of internet in the modern society, it is the duty to protect its users' security and welfare, particularly in light of the skyrocketing risks posed by explicit and violent content. In this paper, we describe a complete method for improving web safety through real-time violence detection in text, photos and videos using machine learning methods, more specifically deep learning Convolutional Neural Networks (CNNs). By performing all processing intensive tasks in the Cloud, and storing the data in a database, as and when the analysis is completed for any website, we can achieve an improved user experience by not just completing all the necessary detection processes at a lower time frame, but also reduce the load on the user's local system. Machine learning models can be used to handle new explicit data with improvement and expansion of training datasets, making sure that the relevance of the system and ability to change in an online dynamic environment.

In conclusion, the effectiveness of utilizing machine learning to improve web safety is demonstrated by the suggested real-time violence detection system. Incorporating mainstream technologies such as Cloud Computing and Bigdata, we consistently achieved an accuracy of 67% over thousands of trials at a time. Moreover, the proposed system architecture achieves all this in under 2.5 seconds at its best-case condition, i.e. results stored in the database. Thus, we have made great progress towards fostering a safer digital environment for all users by fusing technology innovation with ethical considerations.

REFERENCES

- [1] Hidalgo, José María Gómez, Enrique Puertas Sanz, Francisco Carrero García, and Manuel De Buenaga Rodríguez. "Web content filtering." *Advances in computers* 76 (2009): 257-306.
- [2] Gillespie, Tarleton. "Content moderation, AI, and the question of scale." *Big Data & Society* 7, no. 2 (2020): 2053951720943234.
- [3] Lee, Pui Y., Siu C. Hui, and Alvis Cheuk M. Fong. "Neural networks for web content filtering." *IEEE intelligent systems* 17, no. 5 (2002): 48-57.
- [4] Gorwa, Robert, Reuben Binns, and Christian Katzenbach. "Algorithmic content moderation: Technical and political challenges in the automation of platform governance." *Big Data & Society* 7, no. 1 (2020): 2053951719897945.
- [5] Deniz, Oscar, Ismael Serrano, Gloria Bueno, and Tae-Kyun Kim. "Fast violence detection in video." In *2014 international conference on computer vision theory and applications (VISAPP)*, vol. 2, pp. 478-485. IEEE, 2014.

- [6] De Souza, Fillipe DM, Guillermo C. Chavez, Eduardo A. do Valle Jr, and Arnaldo de A. Araújo. "Violence detection in video using spatio-temporal features." In *2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images*, pp. 224-230. IEEE, 2010.
- [7] Ramzan, Muhammad, Adnan Abid, Hikmat Ullah Khan, Shahid Mahmood Awan, Amina Ismail, Muzamil Ahmed, Mahwish Ilyas, and Ahsan Mahmood. "A review on state-of-the-art violence detection techniques." *IEEE Access* 7 (2019): 107560-107575.
- [8] Khan, Samee Ullah, Ijaz Ul Haq, Seungmin Rho, Sung Wook Baik, and Mi Young Lee. "Cover the violence: A novel Deep-Learning-Based approach towards violence-detection in movies." *Applied Sciences* 9, no. 22 (2019): 4963.
- [9] Sumon, Shakil Ahmed, Raihan Goni, Niyaz Bin Hashem, Tanzil Shahria, and Rashedur M. Rahman. "Violence detection by pretrained modules with different deep learning approaches." *Vietnam Journal of Computer Science* 7, no. 01 (2020): 19-40.
- [10] Wang, Pin, Peng Wang, and En Fan. "Violence detection and face recognition based on deep learning." *Pattern Recognition Letters* 142 (2021): 20-24.
- [11] Sernani, Paolo, Nicola Falcionelli, Selene Tomassini, Paolo Contardo, and Aldo Franco Dragoni. "Deep learning for automatic violence detection: Tests on the AIRTLab dataset." *IEEE Access* 9 (2021): 160580-160595.
- [12] Gu, Jiuxiang, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu et al. "Recent advances in convolutional neural networks." *Pattern recognition* 77 (2018): 354-377.
- [13] Jing, Jin, Abdelsalam Sumi Helal, and Ahmed Elmagarmid. "Client-server computing in mobile environments." *ACM computing surveys (CSUR)* 31, no. 2 (1999): 117-157.
- [14] Kumar, Santosh. "A review on client-server based applications and research opportunity." *International Journal of Recent Scientific Research* 10, no. 7 (2019): 33857-3386.
- [15] Lewandowski, Scott M. "Frameworks for component-based client/server computing." *ACM Computing Surveys (CSUR)* 30, no. 1 (1998): 3-27.
- [16] Gupta, Bhumij, and M. P. Vani. "An overview of web sockets: The future of real-time communication." *Int. Res. J. Eng. Technol. IRJET* 5, no. 12 (2018).
- [17] Qveflander, Nikolai. "Pushing real time data usingHTML5 Web Sockets." Digitala Vetenskapliga Arkivet (2010).
- [18] Hormozi, Elham, Hadi Hormozi, Mohammad Kazem Akbari, and Morteza Sargolzaei Javan. "Using of machine learning into cloud environment (a survey): managing and scheduling of resources in cloud systems." In *2012 Seventh International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pp. 363-368. IEEE, 2012.
- [19] Pop, Daniel. "Machine learning and cloud computing: Survey of distributed and saas solutions." *arXiv preprint arXiv:1603.08767* (2016).
- [20] Islam, Tariqul, and Dakshnamoorthy Manivannan. "Predicting application failure in cloud: A machine learning approach." In *2017 IEEE International Conference on Cognitive Computing (ICCC)*, pp. 24-31. IEEE, 2017.
- [21] Wu, Dazhong, Connor Jennings, Janis Terpenney, and Soundar Kumara. "Cloud-based machine learning for predictive analytics: Tool wear prediction in milling." In *2016 IEEE International Conference on Big Data (Big Data)*, pp. 2062-2069. IEEE, 2016.
- [22] Gongane, V.U., Munot, M.V. & Anuse, A.D. Detection and moderation of detrimental content on social media platforms: current status and future directions. *Soc. Netw. Anal. Min.* 12, 129 (2022). <https://doi.org/10.1007/s13278-022-00951-3>
- [23] P. Hajjibabae et al., "Offensive Language Detection on Social Media Based on Text Classification," 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2022, pp. 0092-0098, doi: 10.1109/CCWC54503.2022.9720804.
- [24] Ali Qamar Bhatti, Muhammad Umer, Syed Hasan Adil, Mansoor Ebrahim, Daniyal Nawaz, Faizan Ahmed, "Explicit Content Detection System: An Approach towards a Safe and Ethical Environment", *Applied Computational Intelligence and Soft Computing*, vol. 2018, Article ID 1463546, 13 pages, 2018. <https://doi.org/10.1155/2018/1463546>
- [25] Devin Soni and Vivek K. Singh. 2018. See No Evil, Hear No Evil: Audio-Visual-Textual Cyberbullying Detection. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 164 (November 2018), 26 pages. <https://doi.org/10.1145/3274433>
- [26] S. Parime and V. Suri, "Cyberbullying detection and prevention: Data mining and psychological perspective," 2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014], Nagercoil, India, 2014, pp. 1541-1547, doi: 10.1109/ICCPCT.2014.7054943.