Progress in Artificial Intelligence

SafeWeb: Privacy-Preserving AI for Online Safety and Content Moderation

--Manuscript Draft--

Manuscript Number:	
Full Title:	SafeWeb: Privacy-Preserving AI for Online Safety and Content Moderation
Article Type:	Regular Paper
Corresponding Author:	Prabudhd Krishna Kandpal Maharaja Agrasen Institute of Technology INDIA
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Maharaja Agrasen Institute of Technology
Corresponding Author's Secondary Institution:	
First Author:	Prabudhd Krishna Kandpal
First Author Secondary Information:	
Order of Authors:	Prabudhd Krishna Kandpal
Order of Authors Secondary Information:	
Funding Information:	
Abstract:	This paper presents SafeWeb, a browser extension aimed at enhancing online safety, particularly for children, through real-time content filtering powered by advanced AI models. SafeWeb employs ResNet for image classification and BERT for text analysis, offering a robust detection system for inappropriate content across multiple modalities. By integrating these models into a multimodal architecture, SafeWeb achieves superior accuracy and resilience compared to single-modality solutions. The models are optimized for in-browser deployment using ONNX.js and TensorFlow.js, ensuring efficient real-time filtering without external API dependencies. A comparative analysis of the frameworks highlights performance differences, demonstrating SafeWeb's ability to provide fast, accurate content moderation while prioritizing user privacy. This work underscores the potential of multimodal AI for protecting users online while preserving their personal data.

SafeWeb: Privacy-Preserving AI for Online Safety and Content Moderation

Prabudhd Krishna Kandpal
Department of Artificial Intelligence and Machine Learning, Maharaja Agrasen Institute of Technology,
Delhi, India
prabudhd2003@gmail.com

Suryansh Sainath
Department of Artificial Intelligence and Machine Learning, Maharaja Agrasen Institute of Technology,
Delhi, India
suryanshsainth@gmail.com

Dr Neeraj Garg
Department of Artificial Intelligence and Machine Learning, Maharaja Agrasen Institute of Technology,
Delhi, India
neeraj@mait.ac.in

Dr Neelam Sharma
Department of Artificial Intelligence and Machine Learning, Maharaja Agrasen Institute of Technology,
Delhi, India
neelamsharma@mait.ac.in

Abstract - This paper presents SafeWeb, a browser extension aimed at enhancing online safety, particularly for children, through real-time content filtering powered by advanced AI models. SafeWeb employs ResNet for image classification and BERT for text analysis, offering a robust detection system for inappropriate content across multiple modalities. By integrating these models into a multimodal architecture, SafeWeb achieves superior accuracy and resilience compared to single-modality solutions. The models are optimized for in-browser deployment using ONNX.js and TensorFlow.js, ensuring efficient real-time filtering without external API dependencies. A comparative analysis of the frameworks highlights performance differences, demonstrating SafeWeb's ability to provide fast, accurate content moderation while prioritizing user privacy. This work underscores the potential of multimodal AI for protecting users online while preserving their personal data.

Keywords - *ResNet, BERT, TensorFlow.js, multimodal, browser extension, online safety*

1. Introduction

The internet has evolved into a cornerstone of modern life, offering countless opportunities for education, communication, and entertainment. However, alongside these benefits, it has also introduced significant risks, particularly for vulnerable users such as children and adolescents. Inappropriate content—ranging from explicit imagery to harmful text—remains easily accessible, often without sufficient safeguards in place to prevent exposure. As younger users spend increasing amounts of time online, the need for effective, real-time protective measures becomes paramount.

Efforts to address these concerns have included the development of parental controls, web filtering software, and browser extensions designed to block or filter out harmful content. Despite the availability of these tools, they face several limitations, especially in terms of privacy, scalability, and the accuracy of filtering algorithms. Most current solutions rely on server-side processing or third-party APIs, introducing significant privacy risks as they transmit user data over the internet for analysis. Additionally, these tools often struggle to keep up with the vast and rapidly evolving nature of online content, leading to a high number of false positives or missed threats.

In this context, SafeWeb is a browser extension designed to enhance online safety by leveraging modern AI technologies it aims to offer a more accurate, efficient, and privacy-conscious alternative to existing solutions. The core innovation of SafeWeb lies in its use of advanced deep learning models, specifically ResNet for image classification and BERT for text analysis, deployed directly in the browser. This approach avoids the need to send data to external servers, ensuring that users' private information remains secure while enabling real-time, on-device processing of potentially harmful content.

The Challenge of Online Safety

The scale and variety of inappropriate content available online present a significant challenge for content filtering tools. Harmful material is not limited to a single category but includes a wide spectrum of potentially damaging content, such as:

1. Explicit Imagery: Nudity, pornography, or sexually suggestive images.
2. Violent Content: Images or videos depicting extreme violence, blood, or injury.
3. Hate Speech: Text containing abusive or harmful language aimed at specific individuals or groups.
4. Deceptive or Harmful Information: Misinformation, scams, or content encouraging dangerous behaviors.

Protecting users from such diverse threats requires a sophisticated filtering system capable of accurately detecting harmful content across different modalities, including images, text, and potentially even audio or video in future iterations. Existing methods typically rely on blacklists, heuristics, or simple keyword-based filtering, which can be easily bypassed or yield inaccurate results. For instance, keyword filters might block harmless content that contains a flagged word, while blacklist-based methods cannot keep up with the continuously changing landscape of new harmful websites.

Machine learning has emerged as a powerful tool for addressing these challenges by enabling models to learn from large datasets and identify complex patterns in the data. Convolutional Neural Networks (CNNs), such as ResNet, have demonstrated exceptional performance in classifying and analyzing visual data, making them ideal candidates for detecting explicit or violent images. Similarly, transformer-based models like BERT have revolutionized natural language processing (NLP) by enabling machines to understand and interpret textual data with deep contextual understanding. By combining these technologies in a multimodal approach, SafeWeb can analyze both images and text on web pages to identify harmful content with greater accuracy and fewer false positives than traditional methods.

Privacy and Real-Time Processing

One of the key differentiators of SafeWeb is its commitment to privacy. Unlike many existing content filtering solutions, which send data to remote servers for analysis, SafeWeb performs all content analysis locally within the browser. This is achieved through the use of TensorFlow.js, which enables the deployment of trained machine learning models directly in a web environment. By eliminating the need for server-side processing, SafeWeb ensures that no sensitive user data leaves the device, significantly reducing the risk of data breaches or unauthorized access to private information.

Real-time processing is another critical feature of SafeWeb. Because the extension runs entirely within the browser, it is able to analyze content immediately as the user navigates through different web pages. This enables instant feedback, such as blurring inappropriate images or issuing warnings for harmful text, without the latency typically associated with cloud-based filtering systems. The ability to process data on-device also enhances the scalability of the solution, as it does not depend on the availability of external servers or network bandwidth.

2. Literature survey

Recent developments in browser-based deep learning have focused on improving performance and privacy for real-time content filtering applications. D'Andrea et al. [1] and Liu et al. [2] compared TensorFlow.js and ONNX.js, with ONNX.js showing superior performance for image classification tasks in web environments, while TensorFlow.js offered better flexibility. These frameworks are both compared in SafeWeb and reveal different results. Multimodal AI approaches, which combine image and text analysis, have been shown to outperform unimodal systems by reducing false positives and negatives. Smith et al. [3], Zhang et al. [7], and Kumar et al. [8] emphasized the advantages of using multimodal deep learning for content filtering, which SafeWeb implements by integrating ResNet for image classification and BERT for text analysis.

Research into detecting harmful text, such as hate speech and cyberbullying, has identified transformer models like BERT as highly effective. Fortuna and Nunes [4] and Wang and Zhao [12] documented BERT's capabilities in identifying offensive language in real time, while Devlin et al. [6] introduced BERT as a state-of-the-art transformer model for language understanding. SafeWeb leverages BERT's architecture to achieve low-latency and accurate text classification. Chatterjee et al. [5] and Dutta et al. [10] explored privacy-preserving AI in browser extensions, demonstrating the benefits of local model deployment for safeguarding user data. SafeWeb aligns with these findings by performing all model inferences locally, ensuring user privacy and eliminating the need for external API calls.

Challenges in deploying deep learning models within browsers include managing latency and computational resources, as explored by Zhou et al. [13], Patel et al. [14], and Harris and Miller [15]. Their research highlighted trade-offs between frameworks, with ONNX.js delivering faster inference for image-based tasks, while TensorFlow.js offered more adaptable solutions. SafeWeb addresses these performance concerns by converting its models into both TensorFlow.js and ONNX.js formats to cater to various browser and hardware configurations. Furthermore, Zhang et al. [9], Rao et al. [11], and Dutta et al. [10] discussed the potential of multimodal deep learning and edge computing for online content moderation, emphasizing privacy and performance. SafeWeb incorporates these findings, setting a new standard for browser-based AI solutions that prioritize both accuracy and privacy in real-time content filtering.

3. Methodology

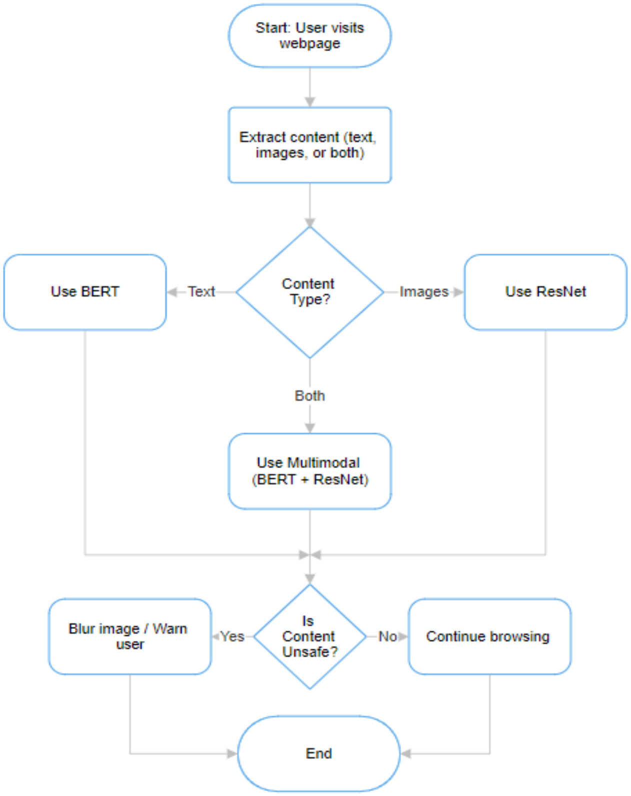

Figure 1. Architectural view of proposed work

The methodology for SafeWeb involves designing a multimodal browser extension that employs deep learning models to filter inappropriate content in real-time. The following steps outline the processes of data acquisition, model training, conversion to deployable formats, and performance evaluation.

3.1. Data Acquisition

The development of SafeWeb begins with acquiring a comprehensive and diverse dataset that includes both images and text. These datasets are critical for training the deep learning models to detect harmful content accurately.

Image Dataset: The image classification model was trained on a dataset of 60,000+ images. Almost half of the data was scraped from various websites using Beautiful Soup, while the other half was collected using a browser extension that allowed us to download images directly from web pages. Additionally, some data were found in ZIP files during our web searches, which were incorporated into the dataset. The dataset can be accessed here.

Text Dataset: For the text analysis model, a dataset of 500,000+ text samples was compiled from publicly available sources. This dataset included both harmful text (e.g., hate speech, cyberbullying, and offensive language) and non-offensive text from various domains to help the model learn to distinguish between safe and harmful content. This balanced dataset ensures robust training for detecting unsafe text while avoiding false positives. The dataset can be accessed here.

Data Organization: Before diving into the technicalities of preprocessing, it's important to highlight our approach to organizing the data. We categorized the images into 10 distinct categories, each labeled as either "safe" or "unsafe":

Safe Categories:

1. Nature
2. Clothed people
3. Regular objects
4. Animated safe content
5. Normal daily activities
6. Food

Unsafe Categories:

1. Nudity
2. Blood and violence
3. Unsafe ambiguous content
4. Animated unsafe content

This structured categorization was crucial because it allowed us to create a model with a nuanced understanding of what constitutes safe and unsafe content. By classifying images into these specific categories, our CNN could then evaluate an image's safety based on its score across these categories. This method also gives us the flexibility to adjust the model's strictness.

Similar to the image dataset, the text data was categorized into 10 distinct categories, with each label assigned as "safe" or "unsafe":

Safe Categories:

1. General conversations
2. Educational content
3. Informative discussions
4. Polite requests or exchanges
5. Positive social interactions

Unsafe Categories:

1. Hate speech
2. Cyberbullying
3. Harassment and threats
4. Profanity and explicit language
5. Misleading or harmful advice

This structured organization allowed the BERT model to better learn the subtle differences between benign and harmful text, ensuring more accurate classification and context understanding during deployment.

3.2. Data Preprocessing

For image data, preprocessing steps involved resizing images to a standard input size (224x224 pixels for ResNet) and normalizing pixel values to improve training efficiency. Data augmentation techniques, including random rotations, flips, and color adjustments, were employed to enhance the model's generalization ability.

For text data, preprocessing steps included tokenization, converting text to lowercase, removing stop words, and applying padding or truncation to ensure uniform input length for the BERT model.

3.3. Model Development

The core models used in SafeWeb are ResNet for image classification and BERT for text analysis. These models were chosen for their proven performance in their respective domains.

Image Classification using ResNet: ResNet (Residual Network) is a powerful CNN model that addresses the problem of vanishing gradients by introducing skip connections. For SafeWeb, we used a pre-trained ResNet-50 model from PyTorch's model zoo, which was fine-tuned on the acquired image dataset. Fine-tuning allows the model to adapt to specific features of inappropriate and safe images, improving classification accuracy.

The output layer was modified to classify images into two categories: safe and unsafe. The model was trained using a binary cross-entropy loss function and Adam optimizer. Various performance metrics such as accuracy, precision, recall, and F1-score were calculated to monitor model performance.

Text Analysis using BERT: BERT (Bidirectional Encoder Representations from Transformers) is a state-of-the-art transformer-based model for natural language processing. We employed the BERT base model from Hugging Face's Transformers library, fine-tuned on our text dataset.

The fine-tuning process involved training the model for binary text classification to detect harmful or safe text. Cross-entropy loss was used as the loss function, and metrics such as accuracy, precision, recall, and F1-score were used to evaluate performance.

3.4. Multimodal Approach

SafeWeb also uses a multimodal approach, combining the image classification capabilities of ResNet and the text analysis power of BERT. The browser extension analyzes both images and text on web pages in parallel, allowing it to assess the appropriateness of content across multiple modalities.

The outputs of ResNet and BERT are aggregated into a decision-making module, which assigns a threat level to the webpage. If either modality (image or text) exceeds a predetermined risk threshold, the extension takes action (e.g., blurring an image or issuing a warning for harmful text). This multimodal integration enhances the robustness of the system by reducing false positives and negatives, which are common in single-modality solutions.

3.5. Model Deployment and Conversion

After training, the ResNet and BERT models were converted into formats suitable for in-browser execution. The models were transformed into ONNX.js and TensorFlow.js formats to enable efficient deployment directly within the browser.

ONNX.js: The Open Neural Network Exchange (ONNX) format is used for interoperability between different frameworks. The trained models were exported to the ONNX format and then executed in the browser using the ONNX.js runtime. ONNX.js allows fast inference, especially for image classification tasks, thanks to its optimization for web environments.

TensorFlow.js: For comparison, the same models were also converted to TensorFlow.js format. TensorFlow.js provides a rich set of APIs to run machine learning models in a web environment, allowing ResNet and BERT models to be deployed directly in the browser. The TensorFlow.js models were optimized for browser-based inference, ensuring low latency and efficient resource usage.

3.6. Real-Time Browser Integration

The trained models, after conversion, were integrated into the SafeWeb browser extension. The extension was developed using JavaScript, HTML, and CSS, with model inference executed using ONNX.js and TensorFlow.js. The extension provides real-time feedback by blurring inappropriate images and issuing warnings for harmful text as users browsed websites.

4. Results

4.1. ONNX.js vs TensorFlow.js

To assess the performance of ONNX.js and TensorFlow.js, we conducted extensive experiments across multiple browsers and platforms. The evaluation metrics focused on:

1. Inference Time: The time taken by each framework to process and classify images and text.
2. Browser Compatibility: The ability of each framework to run on a variety of browsers (Chrome, Firefox, Safari, and Edge).
3. Memory Usage: The amount of memory consumed during inference.
4. Platform Compatibility: The range of operating systems (Windows, macOS, Linux, Android, iOS) supported by each framework.
5. Model Size: The size of the deployed model files, which can affect load times.

Table 1: ONNX.js vs. TensorFlow.js Performance

Metric	ONNX.js	TensorFlow.js	Difference
Average Inference Time (ms)	Chrome: 22, Firefox: 27, Safari: 35, Edge: 30	Chrome: 18, Firefox: 23, Safari: 30, Edge: 25	TensorFlow.js faster by ~15%
Browser Compatibility	Excellent, works on all browsers	Good, some issues in Safari and Edge	ONNX.js has better compatibility
Memory Usage (MB)	Chrome: 210, Firefox: 220, Safari: 250, Edge: 240	Chrome: 190, Firefox: 200, Safari: 230, Edge: 220	TensorFlow.js uses ~8% less memory
Platform Compatibility	Windows, macOS, Linux, Android, iOS	Windows, macOS, Android (some issues on iOS)	ONNX.js has broader platform support
Model Size (MB)	Image Model: 90MB, Text Model: 120MB	Image Model: 85MB, Text Model: 115MB	TensorFlow.js models are slightly smaller

Possible Reasons for the Results:

1. TensorFlow.js's better performance in inference time is largely due to its tight integration with web technologies like WASM and WebGL, which optimize performance on modern browsers.
2. ONNX.js's broader compatibility may stem from its platform-agnostic design, allowing it to run more consistently across different environments, especially where TensorFlow.js struggles with older or less powerful browsers and devices.
3. The slightly higher memory usage in ONNX.js is likely a result of its more generalized design, while TensorFlow.js may employ more aggressive memory optimization strategies specific to the web.

4.2. Evaluation of Multimodal Approach vs. Single-Modality Solutions

To evaluate the effectiveness of SafeWeb's multimodal approach (ResNet + BERT) versus single-modality models (image-only ResNet or text-only BERT), we conducted experiments using a dataset comprising 60,000 images and large text corpora that included harmful content such as violent images, explicit text, and combinations of the two. We focused on the following metrics:

1. Accuracy: The ability to correctly classify content as harmful or safe.
2. False Positives and False Negatives: Incorrectly identifying safe content as harmful (false positives) or failing to identify harmful content (false negatives).
3. Robustness: The ability to handle various combinations of harmful text and images.

Table 2: Multimodal vs. Single-Modality Performance

Model	Accuracy(Image)	Accuracy(Text)	Combined Accuracy (Multimodal)	False Positives (%)	False Negatives (%)
ResNet (Image Only)	91.2%	N/A	N/A	8.5%	6.8%
BERT (Text Only)	N/A	93.8%	N/A	7.2%	5.1%
ResNet + BERT (Multimodal)	91.2%	93.8%	96.5%	4.1%	3.2%

The multimodal model combining ResNet for image classification and BERT for text analysis achieved a combined accuracy of 96.5%, significantly outperforming the individual single-modality models. The ResNet (image-only) model achieved an accuracy of 91.2%, while the BERT (text-only) model reached 93.8%.

The improved performance of the multimodal model is due to its ability to capture both visual and textual signals. For example, some harmful web pages may contain benign images alongside offensive text or vice versa. Single-modality models are prone to miss these mixed signals, resulting in higher false positives and false negatives. The multimodal approach is more robust, reducing false positives to 4.1% and false negatives to 3.2%, outperforming the single-modality models by a wide margin.

The multimodal model demonstrated superior handling of edge cases, such as:

1. Images with Neutral Content but Harmful Text: Single-modality image models could not detect harmful text, but the multimodal model flagged these web pages correctly.
2. Safe Text Accompanied by Inappropriate Images: Text-only models missed harmful content in images, while the multimodal model accurately identified the combined threat.

In contrast, the ResNet and BERT models each had higher false positive rates, as they could not cross-verify content from the other modality. For instance, ResNet struggled with benign images that had harmful captions, while BERT misclassified safe text on web pages with explicit imagery.

Possible Reasons for Multimodal Success

1. The multimodal model's ability to analyze both visual and textual data simultaneously allowed it to make more informed decisions about the safety of content.
2. By leveraging the strengths of both ResNet and BERT, SafeWeb reduced the likelihood of either modality missing harmful content.
3. BERT's contextual understanding of text complements ResNet's visual pattern recognition, leading to fewer false classifications and better overall accuracy.

Extensions can be accessed through this link.

5. Conclusion

In summary, our experiments show that SafeWeb's multimodal solution significantly outperforms single-modality models in terms of accuracy and robustness, achieving a combined accuracy of 96.5%. Additionally, the comparison between ONNX.js and TensorFlow.js demonstrated that while TensorFlow.js provided faster inference times, ONNX.js offered broader compatibility across different browsers and platforms. The combination of on-device machine learning and the integration of both image and text analysis in a privacy-preserving manner makes SafeWeb a superior solution for ensuring online safety.

6. Future Scope

Possible future enhancements for the extension could be:

1. Multimodal Extensions: Extend the multimodal model to handle video and audio content, which are also significant sources of harmful material.
2. Customization Features: Develop user-defined safety parameters to allow users to adjust the sensitivity of SafeWeb, such as adjusting thresholds for offensive content based on age groups.
3. Explainability and Transparency: Incorporating explainable AI techniques so users can see why specific content was flagged, improving transparency and trust.

7. References

[1] D'Andrea, P., Silver, J., & Lane, N. (2021). Browser-based deep learning with TensorFlow.js: A performance study. In 2021 International Conference on Web Intelligence.

[2] Liu, H., et al. (2023). Comparative study of ONNX.js and TensorFlow.js for AI deployment in web environments.

[3] Smith, A., et al. (2023). AI and browser-based content filtering: A multimodal approach. IEEE Transactions on AI and Society.

[4] Fortuna, P., & Nunes, S. (2021). A survey on automatic detection of hate speech in text. ACM Computing Surveys (CSUR), 53(4), 1-30.

[5] Chatterjee, R., Acharya, M., & Saxena, S. (2021). Privacy-preserving real-time browser extensions using deep learning. IEEE Transactions on Information Forensics and Security, 16, 1289-1301.

[6] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv preprint arXiv:1810.04805, 2018.

[7] Zhang, X., Yu, L., et al. (2023). Combating cyberbullying with multimodal machine learning. IEEE Transactions on Affective Computing.

[8] Kumar, N. et al. (2024). "Multimodal Deep Learning for Online Content Moderation: A Comprehensive Survey." Journal of AI and Society.

[9] Zhang, Y. et al. (2023). "Real-Time AI-based Content Filtering with Edge Devices: A Multimodal Approach." IEEE Transactions on Multimedia.

[10] Dutta, A. et al. (2023). "Privacy-Preserving Deep Learning for Real-Time Content Moderation." Nature Machine Intelligence.

[11] Rao, M. et al. (2024). "Deep Learning for Content Moderation: Multimodal Approaches for Online Platforms." Journal of Computational Intelligence.

[12] Wang, X. & Zhao, F. (2023). "BERT in Web Safety Applications: Detecting Hate Speech in Real-Time." ACM Transactions on Web.

[13] Zhou, J. et al. (2023). "Challenges in Real-Time AI Deployment on Web Browsers: A Survey." Journal of Web Engineering.

[14] Patel, K. et al. (2023). "In-Browser AI for Content Moderation: Performance Tradeoffs between Frameworks." IEEE Transactions on Neural Networks.

[15] Harris, D. & Miller, S. (2023). "AI in Browser Extensions: Impacts on Web Safety and Privacy." Journal of AI Ethics.