SafeBrowse: Protection from Not Suitable for Work (NSFW) Images

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering

By

Group No. 04

Simar Kaur - Roll No. 16

Ravi Pandey - Roll No. 43

Jasmit Rathod - Roll No. 54

Supervisor: Prof. Aruna Khubalkar

UNIVERSITY OF MUMBAI

SafeBrowse: Protection from Not Suitable for Work (NSFW) Images

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering

By

Group No. 04

Simar Kaur - Roll No. 16
Ravi Pandey - Roll No. 43
Jasmit Rathod - Roll No. 54

Supervisor: Prof. Aruna Khubalkar

Department of Information Technology

The Bombay Salesian Society's
Don Bosco Institute of Technology
Vidyavihar Station Road, Mumbai - 400070

2023-2024

THE BOMBAY SALESIAN SOCIETY'S
DON BOSCO INSTITUTE OF TECHNOLOGY

Vidyavihar Station Road, Mumbai – 400070

Department of Information Technology

CERTIFICATE

This is to certify that the project entitled **"SafeBrowse"** is the bonafide work of

Simar Kaur	**16**
Ravi Pandey	**43**
Jasmit Rathod	**54**

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Undergraduate in Bachelor of Information Technology.**

Date: / 04 / 2024

(Prof. Aruna Khubalkar)
Supervisor

(Prof. Prasad Padalkar)
HOD, IT Department

(Dr. Sudhakar Mande)
Principal

THE BOMBAY SALESIAN SOCIETY'S
DON BOSCO INSTITUTE OF TECHNOLOGY

Vidyavihar Station Road, Mumbai – 400070

Department of Information Technology

Project Approval Report for B.E.

This project report entitled **"SafeBrowse"** by **Simar Kaur (Roll No. 16), Ravi Pandey (Roll No. 43) and Jasmit Rathod (Roll No. 64)** is approved for the degree of **Bachelor of Engineering in Information Technology.**

(Examiner's Name and Signature)

1. Prof. Aruna Khubalkar

2. _____

(Supervisor's Name and Signature)

Prof. Aruna Khubalkar

(Chairman)

Prof. Janhavi Baikerikar

Date: / 04 / 2024
Place: Mumbai

THE BOMBAY SALESIAN SOCIETY'S
DON BOSCO INSTITUTE OF TECHNOLOGY

Vidyavihar Station Road, Mumbai – 400070

Department of Information Technology

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact /source in our submission. We understand that any violation of the above will cause disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Simar Kaur Dua
Roll No. 16

Ravi Pandey
Roll No. 43

Jasmit Rathod
Roll No. 54

Date: / 04 / 2024

Abstract

In today's online world, the increasing presence of Not Suitable for Work (NSFW) content raises concerns about its potential detrimental impact to mental well-being. Amidst these concerns, the research paper introduces "**SafeBrowse**", a novel browser extension meticulously designed to enhance online safety by preventing the display of NSFW images. What sets SafeBrowse apart is its unique approach, departing from conventional methods relying on simplistic keyword lists and image recognition algorithms. The extension employs a sophisticated process, beginning with the extraction of image URLs from web pages. It then utilizes advanced image-to-text models, specifically GIT and BLIP, to generate detailed textual descriptions of images. The all-MiniLM-L6-v2 sentence transformer model is used to calculate the cosine similarity score between the captions generated by GIT and BLIP using a threshold of 0.65 to determine selection of more accurate caption amongst the two. Followed by which, a finely-tuned deBERTa model subsequently classifies captions into categories such as violent, sexually explicit, and neutral, harnessing both visual and textual data to enrich content evaluation. The overall accuracy of our model SafeBrowse is 69.15%.

While initial pilot testing emphasizes the effectiveness of the ensemble approach rather than specific accuracy metrics, the paper outlines a forward-looking agenda, prioritizing enhancements in user experience. Users can access the extension's user interface through a convenient popup window, triggered by clicking on the extension icon. This project is poised to make a significant impact in the online ecosystem by enhancing user safety, reducing unwanted content exposure, and granting individuals greater autonomy over their online experience. With compatibility across multiple web browsers and platforms, it promises to be a valuable tool for a diverse user base.

Keywords - Machine learning, Image-to-text generation , Text Classification , NSFW content filtering , Not Safe For Work Images, Multi-model approach , Browser extension , Ensemble learning, Safe browsing.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

To create a highly effective and adaptable Not Safe For Work (NSFW) image recognition browser extension that empowers users to identify and block explicit or inappropriate content fostering a respectful and safe online environment for all users.

1.2 Scope of the Project

The filtering system is designed to be versatile, making it applicable to multiple websites, providing a robust content moderation solution for home-based systems. It allows users to adapt sensitivity levels and apply manual overrides, giving them the power to hide specific content as needed. This customization is based on image classification within a few key categories, including neutral content, violent content, and sexually explicit material. As such, this system caters to a broad target audience by providing a flexible and user-centric approach to online content moderation in home environments.

Following are the assumptions from various phases of development:

• Requirements Gathering Phase

I. It is assumed that users are aware of the presence of NSFW content on the internet and assume that the definition of NSFW is standard for all.

II. There is no bias in the data such as reporting bias, selection bias, gender bias.

• Requirements Analysis Phase

I. Browser Compatibility: The extension is for web browsers only and is assumed to be compatible with major web browsers, including but not limited to Google Chrome, Mozilla Firefox, and Microsoft Edge.

II. Caption Generation Accuracy: The GIT model is assumed to provide accurate and contextually relevant captions for images to support effective content analysis.

• System Design phase

I. Extension is compatible on all devices & browsers.

II. Response time is 1.5 minutes for processing 9 images.

Following are the limitations faced:

I. Duplicate Images on Different Hosted Links: The extension treats the same image hosted on different URLs as distinct images, potentially leading to duplicated storage and processing. This limitation arises because the extension identifies images based on their URL. A single NSFW image posted in multiple locations may be flagged multiple times, which could affect user experience and storage efficiency.

II. Dependency on Hosted Links: The extension relies on images being hosted on external links. Images embedded in web pages through local file uploads (e.g., using the "file://" protocol) are not within the scope of this project. Consequently, the extension will not process or analyze locally uploaded images, as they are not accessible via external URLs.

III. Incompatibility with <iframe> Tags: If images are deployed within web pages using the <iframe> HTML tag, the extension may encounter limitations in its ability to access and analyze these images. This is because content within iframes is typically isolated from the main page's DOM (Document Object Model), making it challenging for the extension to interact with and analyze the content within the iframe.

IV. No Real-time Updates: The extension's NSFW content detection and filtering capabilities are based on pre-trained models. It does not include a mechanism for real-time updates of these models to adapt to emerging NSFW content trends. Therefore, the extension may not always catch newly emerging NSFW content categories or variations.

V. Performance Impact: Depending on the complexity and volume of images on a webpage, the extension's image processing and caption generation may introduce a performance overhead, potentially affecting page load times. This limitation should be considered when evaluating the extension's suitability for resource-intensive web pages.

VI. Limited Language Support: The extension may have limitations in handling content in languages other than English, as the pre-trained models and caption analysis target English-language content. Content in other languages may not be classified or processed as accurately.

VII .Compatibility with Future Browser Updates: The extension's compatibility with future updates of web browsers may not be guaranteed. Browser updates can introduce changes to APIs and security policies that could impact the extension's functionality and may require updates to maintain compatibility.

VIII. Categorization Accuracy: The accuracy of NSFW content classification into predefined categories may vary depending on the quality and diversity of the training data. Some content may be misclassified or fall into ambiguous categories, and users should exercise caution when relying solely on the categorization feature.

1.3 Current Scenario

Existing browser extensions that aim to filter NSFW content have often faced challenges, including inaccuracies, false positives, and false negatives. These limitations have underscored the need for more sophisticated and precise solutions. A promising approach to address these issues is the integration of a hybrid system that combines tag-based methods with machine learning techniques. By leveraging tags as an initial categorization mechanism and then continuously training machine learning models, the extension can refine its content detection accuracy over time. This dynamic approach not only builds upon existing tagging systems but also allows for adaptation to evolving online content trends, providing users with a more effective and reliable NSFW content filtering experience. Online privacy and data security are becoming increasingly important topics. Users are more cautious about the data

they share, including with browser extensions. The new extension addresses these privacy concerns transparently.

1.4 Need for the Proposed System

The presence of NSFW content in professional and academic settings can cause disruptions and potential harm. It can create uncomfortable workspaces, damage professional reputations, and even lead to disciplinary actions. Continuous exposure to explicit material might lead to desensitization, creating emotional detachment and triggering feelings of shame or lowered self-esteem [1]. The ready accessibility of explicit content online further compounds these challenges, potentially fostering compulsive behavior that adversely affects both mental well-being and interpersonal relationships. In educational settings, it distracts students from their studies and hinders the creation of a conducive learning environment. Therefore, it is crucial to block NSFW content in these settings in order to maintain professionalism, focus, and a respectful atmosphere. In light of the ease of internet accessibility for children, there is an increased risk of them being exposed to inappropriate content that is not suitable for work (NSFW). This exposure can have detrimental effects on a child's development and psychological well-being. It is crucial to implement parental controls and filters to block NSFW content in order to protect children from being exposed to inappropriate material. These measures also empower parents and guardians to effectively manage the digital content that their children are exposed to.
A NSFW (Not Safe For Work) filter for web browsers can serve several important purposes and cater to different user needs:
Protecting Users from Inappropriate Content: The primary purpose is to protect users from encountering explicit or inappropriate content that may be offensive or disturbing.

Parental Control: As our system is for home based systems, NSFW filters are valuable tools for parents who want to restrict their children's access to explicit or age-inappropriate content on the internet. It helps ensure that children are not exposed to content that is not suitable for their age.

Customization and Personalization: NSFW filters often empower users to customize their browsing experience according to their preferences. Users can set the filter to their desired level of strictness, allowing them to define what content is considered explicit or inappropriate based on their personal standards.

Compliance with Regulations: In some regions, there are laws and regulations that require the filtering of explicit content to comply with local standards or protect certain demographics, such as minors. Our proposed solution can help websites and platforms adhere to these regulations.

Privacy and Consent: NSFW filters can also play a role in respecting the privacy and consent of individuals. By allowing users to filter out explicit content, it ensures that individuals have more control over the type of content they consume and engage with.

Filtering User-Generated Content: Many websites and social media platforms rely on user-generated content, which can sometimes include explicit or inappropriate material. NSFW filters can help these platforms automatically identify and filter out such content, creating a safer and more user-friendly environment.

Chapter 2

Review of Literature

2.1 Summary of the investigation in the Published Papers

We studied the following papers and discovered the facts mentioned below:
1) Bicho, Daniel et al. present a DNN-based solution for NSFW image classification from the Arquivo.pt web archive, achieving an impressive 94% accuracy[2]. They employ transfer learning with the Arquivo.pt dataset, which contains historical Portuguese web pages and images. The research signifies a substantial advancement in content moderation for web archiving, although it comes with computationally intensive feature extraction as a challenge.

2) Zhuravlev, Sergey et al. introduce the ChildNet model, utilizing a 21-layer deep neural network with reduced filter size to analyze pixel patterns in digital images, focusing on pixel-based nudity detection[3]. The model demonstrates superior performance compared to classical Convolutional Neural Networks, offering promise for a safer online environment for young users.

3) Smith, John et al. emphasize the power of Convolutional Neural Networks (CNNs) in achieving 97% accuracy for Not Suitable For Work (NSFW) content detection [4], including natural human nudity and explicit illustrations. They also highlight the importance of a comprehensive dataset for effective NSFW image filtering, offering insights into evolving methods.

4) Ahmed, Faraz et al. present an explicit content detection system based on a residual network, which accurately classifies NSFW media content, providing gradations of explicitness. Experimental results show a high accuracy of around 95% [5], contributing to safer digital environments.

5) Lienhart, Rainer & Hauke, Rudolf address the urgent need to protect children from adult content online, citing alarming statistics. Their research assesses the use of probabilistic Latent Semantic Analysis (pLSA) to detect adult images, achieving a robust 92.7% correct positive rate with a low 1.9% false positive rate [6]. Even when applied to grayscale images exclusively, the method maintains a commendable 90.8% correct positive rate and a 2% false positive rate. This research presents a promising solution for shielding young internet users from explicit content.

6) Vincent delves into the vital need for automated content moderation in web and mobile applications flooded with user-generated image uploads, emphasizing the detection of NSFW (not safe for work) content, which is crucial for maintaining a secure online environment. The author highlights the impracticality of relying on human moderators for each image upload and underscores the role of Computer Vision, a facet of Artificial Intelligence, in automatically and accurately classifying NSFW content. This post provides insights into utilizing the PixLab API [7] to detect

and filter unwanted content, including GIFs, while offering actions based on the PixLab score, such as blurring or deleting the image.

7) In the domain of client-side content moderation, NSFW JS, an open-source JavaScript library, excels in identifying potentially indecent images directly within the user's browser. It employs machine learning models to achieve high accuracy (about 90% with small models and 93% with mid-sized ones [8]) while prioritizing user privacy by conducting real-time analysis without external data transmission. Although NSFW JS has limitations, it represents a promising advancement in the pursuit of efficient and privacy-conscious content filtering, as outlined in the literature.

8) The "NSFW Detector" provided by DeepAI.org showcases a significant contribution to content moderation and indecency detection [9]. Leveraging advanced machine learning techniques, this model excels in identifying not safe for work (NSFW) content in various forms, such as images and text. DeepAI.org's innovation demonstrates the evolving capabilities of AI and machine learning in content moderation, offering valuable insights and tools to enhance the ability to filter inappropriate content in various digital contexts, as acknowledged in the relevant literature.

2.2 Comparison with existing systems

Table 1. Comparison of Existing systems and SafeBrowse

Model	Focus	Method	Accuracy	Limitations
DNN-based solution by Bicho	Web archiving	Transfer learning with Arquivo.pt dataset	94%	Computationally expensive feature extraction
ChildNet by Zhuravlev	Pixel-based nudity detection	21-layer deep neural network with reduced filter size	Superior to classical CNNs	Limited to pixel-based features
CNNs based architecture by Smith	NSFW content detection	Convolutional Neural Networks	97%	May struggle with grayscale images
pLSA by Lienhart & Hauke	Protecting children online	Probabilistic Latent Semantic Analysis	92.7% correct positive rate, 1.9% false positive rate	Lower accuracy compared to some CNN models
SafeBrowse (Our model)	Enhance browsing experience by blocking NSFW images	Blocking images dynamically by extracting them from URLs	Combined accuracy of 69.15%	Real time user feedback not considered

2.3 Existing Modules

2.3.1 Description:

A. GIT - Generative-Image-to-Text

Fig 1: Example of GIT model [10]

Dataset: COCO Train 2017: 118,000 images along with their corresponding captions.
COCO Val 2017: The validation set consisted of around 5,000 images with captions.
COCO Test 2017: The test set contained approximately 41,000 images.

Microsoft's git-large-coco has emerged as a prominent model for Image-to-Text Transformation, showcasing remarkable precision in image captioning tasks. GIT is a Transformer decoder conditioned on both CLIP image tokens and text tokens[10]. This model leverages the GIT architecture and undergoes pretraining on an extensive dataset of 14 million images, followed by further fine-tuning on the well-established COCO benchmark. Consequently, this rigorous training regimen culminates in an impressive CIDEr score of 138.5 on COCO captioning tasks, surpassing the performance of smaller models that typically attain BLEU-4 scores of around 35 [11]. It is worth noting that git-large-coco's BLEU@4 score surpasses 42 [11], further reinforcing its competitive advantage in generating accurate and coherent captions. This innovative approach empowers git-large-coco to extract intricate visual details and relationships within images, subsequently translating them into comprehensive and nuanced textual descriptions.

B. BLIP - Bootstrapping Language-Image Pre-training for Unified Vision

BLIP, which stands for Bootstrapping Language-Image Pre-training is a state-of-the-art model for image captioning tasks. Its exceptional performance is evident in its impressive CIDEr score of 136.7 [13], indicating accurate and detailed caption generation. Additionally, it achieves a high BLEU@4 score of 40.4 [13], demonstrating fluency and coherence in its generated captions. The success of BLIP can be attributed to its architecture and training process. This model utilizes the powerful ViT-L (Vision Transformer Large) backbone to extract comprehensive visual features from images.

Fig 2: Example of BLIP model [12]

Subsequently, it employs a unique bootstrapping approach, where it generates synthetic captions and eliminates inaccurate ones using a specialized "captioner" module. This iterative process enables BLIP to develop a robust understanding of the relationship between visual and textual data. The BLIP Model uses ViT-L/16 as a vision backbone.
Pretrain dataset: COCO+VG+CC+SBU+LAION (129M images)
Metrics: Image Captioning- Fine Tune: B@4 (COCO): 40.4, CIDEr (COCO): 136.7
Image Captioning- Zero Shot: CIDEr (NoCaps): 113.2, SPICE: 14.8 [13]
Moreover, BLIP's training is based on the well established COCO dataset, comprising images paired with relevant captions. This extensive dataset allows the model to learn from a diverse range of scenarios and objects, thereby enhancing its ability to generalize to unseen images.

C. all-MiniLM-L6-v2 - Sentence Transformers

To effectively evaluate the similarity between captions produced by the BLIP and GIT models, a robust and efficient method is essential. Our approach employs allMiniLM-L6-v2, a streamlined version of Microsoft's MiniLM-L12 model. Despite its small 80MB size, it exhibits impressive performance, encoding 14,200 sentences per second and achieving an average sentence score of 58.80 [14]. The model can process inputs of up to 256 word pieces, benefiting from fine-tuning on a large dataset of over a billion sentences from sources like Reddit comments, S2ORC, WikiAnswers, and COCO captions. This comprehensive training enhances the model's ability to understand semantic relationships in captions. Our comparison between captions emphasizes cosine similarity, assessing each pair's resemblance by calculating the cosine similarity in batch pairs and comparing the results with the ground truth using cross-entropy loss. This meticulous analysis offers quantitative insights into the resemblance of captions from BLIP and GIT models, enabling a thorough evaluation of their generation capabilities. Leveraging the efficiency and precision of the allMiniLM model enhances our understanding of caption semantics, driving advancements in this field. Our goal is to contribute to caption generation progress by diligently evaluating model similarities in a dependable and unbiased manner.

D. DeBERTa:

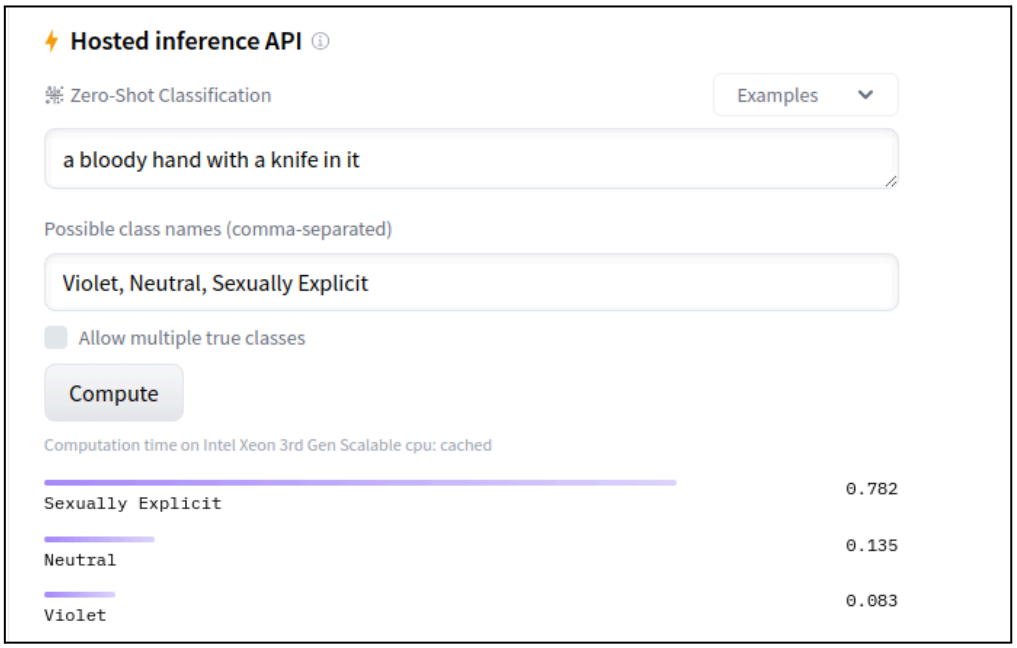


Fig 3: Example of DeBERTa [15]

Decoding-enhanced BERT with Disentangled Attention DeBERTa represents a groundbreaking advancement in natural language processing, distinguished by its innovative architectural design and exceptional performance across various tasks. Unlike traditional models, DeBERTa dissects words into their fundamental meaning and positional information, fostering a deeper understanding of the intricacies of language. Through its enhanced decoder mechanism, which is trained with absolute positional cues, DeBERTa accurately predicts missing words and navigates complex sentence structures.

Outperforming renowned models such as BERT and RoBERTa, DeBERTa achieves a record breaking average score of 90.00% on the GLUE benchmark[16]. It demonstrates superior performance in answering tasks on the Stanford Question Answering Dataset(SQuAD) benchmark and excels in named entity recognition tasks on the CoNLL-2003 NER benchmark. Remarkably, despite being trained on a relatively modest 78GB dataset, DeBERTa surpasses models trained on larger datasets, making it a compelling choice for real-world applications with limited data resources.

In summary, DeBERTa's innovative architecture, coupled with its exceptional performance and data efficiency, positions it as a leading model in NLP, with the potential to revolutionize language understanding and interaction across diverse applications.

Comparison Chart of Selected Model with other similar model

I. Comparison of Image to Text models:

Table 2 : GIT vs LSTM

ASPECT	GIT	LSTM
Full Form	GenerativeImage2Text Model	Long Short Term Memory Model
BLEU score	Higher BLEU scores (e.g., 0.7) indicating better caption quality	Lower BLEU scores (e.g., 0.5) indicating potentially lower caption quality
Model Architecture	Combines CNNs and Transformers (e.g., GPT)	Uses CNNs for image features and LSTM cells
Performance	Outperforms LSTM in terms of caption quality, coherence, and relevance	May produce captions with less context and occasional issues like "hallucinations"
Training	Computationally intensive; requires large data and significant resources	Easier to train, requires fewer resources
Handling Complex Context	Excels at capturing complex visual and contextual relationships	May struggle with long-range dependencies and complex context
Adaptability and Fine Tuning	Can be fine-tuned for domain-specific tasks	Can also be fine-tuned, but may have limitations
Application and Usage	Generally higher accuracy in generating contextually relevant captions	May have lower accuracy in generating contextually accurate captions
Accuracy	Approximately 85-90% [11] accuracy in generating contextually relevant captions	Approx. 70-80% [17] accuracy in generating contextually relevant captions

II. Comparison of Text Classification models:

BERT, RoBERTa, and DeBERTa [18] are all transformer-based language models that have been pre-trained on large amounts of text data to perform a variety of NLP tasks.

Table 3: BERT vs RoBERT vs DeBERTa

ASPECT	BERT	RoBERTa	DeBERTa
Full Form	Bidirectional Encoder Representations from Transformers	Robustly Optimized BERT Pretraining Approach	Decoding-enhanced BERT with Disentangled Attention

ASPECT	BERT	RoBERTa	DeBERTa
Technique Used	Bidirectional Transformer	Masked Language Modeling uses dynamic masking instead of static masking.	Dynamic Embeddings + RoBERTa-like Transformer
Accuracy	Varies (pre-training) and fine-tuning	Competitive with state-of-the-art	Competitive with state-of-the-art
Training DataSet	BookCorpus, English Wikipedia	BookCorpus, English Wikipedia. RoBERTa was trained on 10x more data than BERT.	BookCorpus, English Wikipedia
Pre-training	BERT uses bidirectional context prediction	RoBERTa focuses solely on masked language modeling	DeBERTa utilizes dynamic embeddings and a similar architecture to RoBERTa
Attention Masks	BERT uses fixed patterns for attention masks	RoBERTa uses dynamic patterns for attention masks	DeBERTa uses dynamic patterns for attention masks
Token Processing	BERT uses WordPiece tokenization	RoBERTa also uses BPE for subword tokenization	It uses Byte-Pair Encoding (BPE) for subword tokenization
Training Techniques	BERT uses Next Sentence Prediction (NSP) and masked language modeling (MLM)	RoBERTa emphasizes a larger batch size and longer training duration	DeBERTa uses a similar MLM approach to RoBERTa and incorporates dynamic embeddings
Notable Features	BERT introduced the concept of transformer-based pretraining for NLP	RoBERTa refined the pretraining of BERT by using larger batch sizes and training for more epochs	DeBERTa improved upon BERT by using dynamic embeddings and a similar architecture to RoBERTa
Achievements	BERT set the foundation for many subsequent NLP models and achieved competitive performance	RoBERTa achieved state-of-the-art performance on various NLP benchmarks with its modifications to BERT's pretraining	DeBERTa achieved competitive performance with state-of-the-art models while emphasizing dynamic embeddings
Performance	BERT offers competitive performance, and its performance can be further enhanced with careful fine-tuning.	Dynamic masking allows RoBERTa to achieve better performance than BERT on several natural language processing tasks.	It improves the BERT and RoBERTa models using two techniques : i. disentangled attention mechanism ii. enhanced mask decoder.

It's essential to consider specific use cases and task requirements when choosing one of these models for a particular NLP application.
In summary, while all three models are transformer-based language models that have been pre-trained on large amounts of text data, they differ in their training data, masking strategies, and additional mechanisms introduced in their architectures.
DeBERTa is a Transformer-based neural language model that improves the BERT and RoBERTa models using two novel techniques: a disentangled attention mechanism and an enhanced mask decoder. It is not based on n-gram, bigram, or trigram models.
The disentangled attention mechanism is where each word is represented using two vectors that encode its content and position, respectively, and the attention weights among words are computed using disentangled matrices on their contents and relative positions. The enhanced mask decoder is used to replace the output softmax layer to predict the masked tokens for model pretraining.
Disentangled attention helps the model to focus on different aspects of the input sequence, while the enhanced mask decoder improves the quality of the masked tokens.
Deberta consists of 48 Transform layers with 1.5 billion parameters.

2.3.2 Algorithm(s) with example

1) GIT (GenerativeImage2Text)-large model

Step 1: Initialize the GIT model which is essentially a Transformer-based architecture comprising multiple layers of self-attention, feed-forward neural networks, and positional encodings.

```
def __init__(self):
        # Initialize the model with transformer-based architecture
        self.transformer = TransformerModel()
```

Code snippet for initialization of the model

Step 2: Tokenize the input text and images to represent them as numerical sequences suitable for model input. This process is fundamental in preparing the data for further processing.

```
def tokenize_data(text, images):

        # Tokenize the text data
        text_tokens = tokenize_text(text)

        # Tokenize and preprocess the image data
        image_tokens = preprocess_images(images)

        return text_tokens, image_tokens
```

Code snippet for tokenization

Step 3: Next, It creates a model input by combining image tokens and text tokens, allowing for the establishment of cross-modal connections. Positional encodings are added to retain spatial information

in image tokens.

```
# Pseudo code for creating model input
def create_model_input(text_tokens, image_tokens):

        # Combine text and image tokens
        model_input = text_tokens + image_tokens

        # Add positional encodings for spatial information
        model_input_with_position = add_positional_encodings(model_input)

        return model_input_with_position
```

Code snippet for creating model input

Step 4: Initialize an empty sequence for the generated text, which will be built incrementally through recurrent iterations.

```
# Pseudo code for initializing the generated text sequence
        def initialize_generated_text():
         generated_text = []              # Initialize an empty list to hold tokens
         return generated_text
```

Code snippet for initialization of generated text sequence

Step 5**:** The model processes the unified input using a multi-head self-attention mechanism, which facilitates capturing dependencies between tokens, followed by feed-forward layers that transform token representations.

```
# Pseudo code for processing the unified input
        processed_input = model.process(model_input)
```

Code snippet for processing the unified input

Step 6: The model employs two distinct attention mechanisms:
Image Attention: Multi-head self-attention enables the model to capture contextual relationships among image tokens, facilitating a comprehensive understanding of visual context.

```
# Pseudo code for applying image attention
        image_attention_output = model.image_attention(processed_input)
```

Code snippet for applying image attention

Step 7: Text Attention: Causal (autoregressive) self-attention is applied to text tokens, ensuring the model generates text sequentially based on preceding tokens.

```
# Pseudo code for applying text attention
        text_attention_output = model.text_attention(processed_input, generated_text)
```

Code snippet for applying text attention

Step 8: Utilizing the processed representations, the model estimates the probability distribution over the vocabulary to predict the next token. This is achieved through a softmax operation.

```
# Pseudo code for estimating the probability over the vocabulary
        vocabulary_distribution = model.predict_next_token(image_attention_output,
        text_attention_output)
```

Code snippet for estimating probability over the vocabulary

Step 9: Employ sampling techniques such as top-k sampling or nucleus sampling to stochastically select the next token, introducing an element of randomness in the generation process.

```
# Pseudo code for sampling technique to select the next token
        next_token = sample_token(vocabulary_distribution)
```

Code snippet for sampling technique

And then append the sampled token to the existing sequence of generated text.

```
# Append the sampled token to the generated text
        generated_text.append(next_token)
```

Code snippet for appending the sampled token to the generated text

Note: The process (from step 5 through 9) is part of an iterative process where the model generates text one word at a time until it reaches a predefined stopping conditions, which can be based on a maximum token length or the generation of a specified end token.

```
        while not stopping_condition(generated_text):
```

Code snippet for stopping condition

The concatenated text sequence represents the textual description of the input image and text tokens.

Step 10: Conduct any necessary post-processing, such as removal of special tokens, handling punctuation, and ensuring the generated text adheres to language constraints.

```
        # Pseudo code for post-processing
        def post_process_generated_text(generated_text):
        # Perform any necessary post-processing, e.g., removing special tokens etc.
        processed_text = post_process(generated_text)
        return processed_text
```

Code snippet for post processing

Step 11: Provide the generated text as the output of the GIT model, often after further processing to refine the final result.

```
# Pseudo code for final output
def generate_output(processed_text):
# Provide the generated text as the output
return processed_text
```

Code snippet for final output

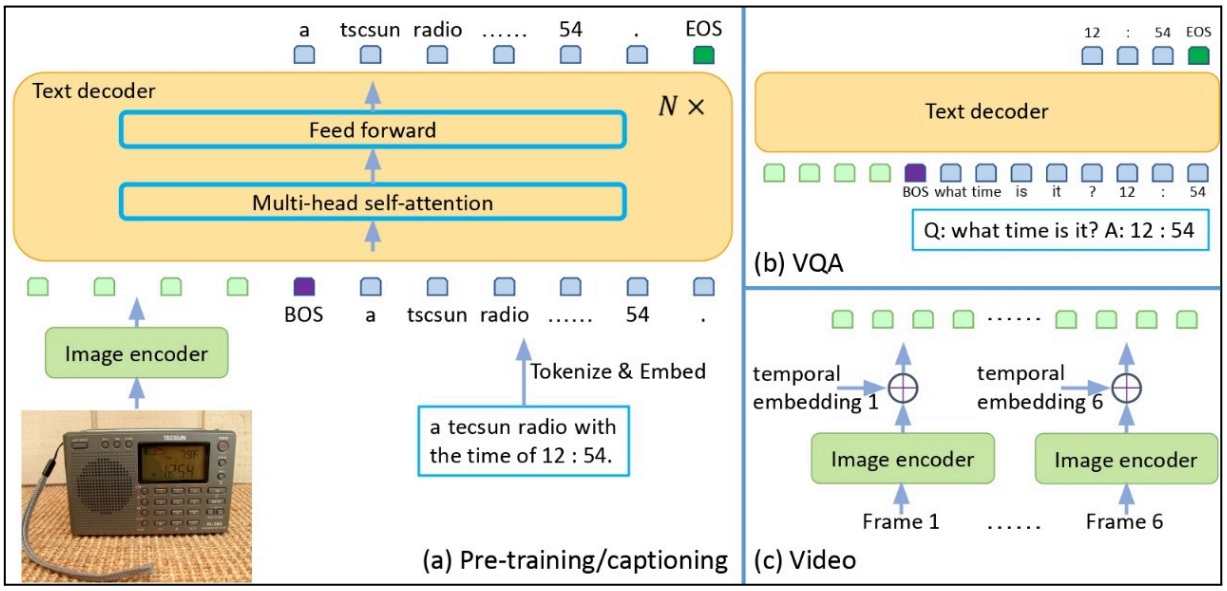


Fig 4: Internal Flow of GIT model [11]

Time Complexity of GIT model:

Input Processing: If the tokenization and processing operations are O(n), this part has a time complexity of O(n).
where, "n" tokens in the input (text and images)

Text Generation: The complexity of this part depends on the number of iterations needed before the stopping condition is met and the complexity of each iteration.
where, k is the number of iterations to generate the text

Post-Processing: This part has a time complexity of O(m).
where, the generated text has "m" tokens.

Overall: It can vary from O(n + k * [complexity of each iteration] + m)

Space Complexity:

Model Size: If the model has "p" parameters, it requires memory proportional to "p".
Input Data: O(n).
where, n is the number of tokens

Generated Text: O(m).
where m is the length of the generated text

Intermediate Data: The space used for intermediate computations during text generation, including attention matrices and hidden states, can also affect space complexity.

Overall: The overall space complexity of the GIT model is quite significant due to the model's size and the amount of data it processes.

2) DeBERTa V3 base Model

Step 1: Initialization

I. Initialize the DeBERTa model with architectural parameters like the number of layers, hidden dimensions, etc.
II. Optionally, load pre-trained weights from a DeBERTa model checkpoint.

```
# Pseudo code for model initialization
class DeBERTaModel:
    def __init__(self, num_layers, hidden_dim, pretrained_weights=None):
        self.num_layers = num_layers
        self.hidden_dim = hidden_dim

        if pretrained_weights:
            self.load_weights(pretrained_weights)
        else:
            self.initialize_weights()

    def initialize_weights(self):
        # Initialize model's weights, embeddings, and parameters here.

    def load_weights(self, pretrained_weights):
        # Load pre-trained weights from a checkpoint file.
```

Code snippet DeBERTa model initialization

Step 2: Tokenization

I. Tokenize the input text into subword tokens using a tokenizer like SentencePiece or WordPiece.
II. Convert the tokens into embedding vectors, including token embeddings, position embeddings, and segment embeddings.

```
# Pseudo code for tokenization and embedding
class DeBERTaModel:
def tokenize_and_embed(self, input_text):
                    subword_tokens = self.tokenizer.tokenize(input_text)
                    embedding_vectors = self.embed_tokens(subword_tokens)
                    return embedding_vectors
```

```
    def embed_tokens(self, tokens):
        # Convert tokens to embedding vectors using token, position, and segment
embeddings.
        return embedding_vectors
```

Code snippet for tokenization and embedding

Step 3: Positional Encoding

I. Add positional embeddings to the token embeddings to capture word positions in the sequence.

```
seudo code for positional encoding
s DeBERTaModel:
ef add_positional_embeddings(self, embedding_vectors):
 # Add positional embeddings to the token embeddings.
 return embeddings_with_positions
```

Code snippet for positional encoding

Step 4: Disentangled Attention Layers

I. DeBERTa uses multiple layers, each comprising the following components:
   A. Disentangled Self-Attention Mechanism
      1. Calculate Query (Q), Key (K), and Value (V) tensors from input embeddings.
      2. Apply disentangled attention masks to Q and K to capture different dependencies.
      3. Compute attention scores as Q * K^T.
      4. Apply a mask (usually to ignore padding tokens).
      5. Calculate weighted sums using attention scores to obtain the attention output.
   B. Layer Normalization and Residual Connection
      1. Apply layer normalization to the attention output.
      2. Add the attention output to the original input as a residual connection.
   C. Feed-Forward Neural Network
      1. Pass the output from the attention mechanism through a feed-forward neural network.
   D. Layer Normalization and Residual Connection
      1. Apply layer normalization to the feed-forward output.
      2. Add the feed-forward output to the output of the attention mechanism as a residual connection.

```
# Pseudo code for a single disentangled attention layer
class DisentangledAttentionLayer:
        def forward(self, input_embeddings):
          # Disentangled self-attention mechanism
          query, key, value = compute_qkv(input_embeddings)
          attention_scores = compute_attention_scores(query, key)
          attention_output = apply_attention(attention_scores, value)

          # Layer normalization and residual connection
```

```
    normalized_output = layer_normalize(attention_output)
    output = input_embeddings + normalized_output
    # Feed-forward neural network
    ff_output = feed_forward(output)

    # Layer normalization and residual connection
    normalized_ff_output = layer_normalize(ff_output)
    output = output + normalized_ff_output
    return output
```

Code snippet for a single disentangled attention layer

Step 5: Layer Stacking

I. Repeat Step 4 for a predefined number of layers, commonly ranging from 12 to 24. This increases the model's ability to capture complex patterns and dependencies.

Step 6: Training

I. Fine-tune the DeBERTa model on a specific NLP task using labeled data and an appropriate loss function.

II. Employ backpropagation and an optimizer like Adam to update the model's parameters during training.

```
def train_model(model, train_data, optimizer, loss_function, num_epochs):
    for epoch in range(num_epochs):
        for batch in train_data:
            # Forward pass
            predictions = model(batch)

            # Compute the loss
            loss = loss_function(predictions, batch)

            # Backpropagation
            optimizer.zero_grad()
            loss.backward()
            optimizer.step()
```

Code snippet for training the model

Step 7: Inference

I. Use the trained DeBERTa model to make predictions on new text data for various NLP applications.

Chapter 3

Analysis and Design

3.1 Methodology

In this methodology, we outline a structured approach that amalgamates Agile principles with a well-defined project framework to ensure the successful development of our browser extension, 'SafeBrowse:Protection from NSFW Images,'.

Requirement Gathering and Analysis:
i. Project Goals: The objectives of the browser extension were clearly articulated, encompassing NSFW content filtering and image caption generation.
ii. User Research: User needs and preferences were thoroughly understood, taking into account their content filtering preferences and desired image captioning features, aligning with Agile's user-centric principles.
iii. Content Categories: Specific NSFW content categories that the extension should detect and allow users to filter were identified, following research into common NSFW content types, guided by Agile's focus on user needs.
Planning and Design:
i. User Story Mapping: We created user stories and user story maps to outline the user journey, specifying how users would interact with the extension for content filtering and image captioning, and we also thought about the rough logic of how it would be done.
ii. Wireframing and Prototyping: We then developed wireframes and interactive prototypes to design the user interface for the extension window, ensuring it was user-friendly and intuitive.
iii. Algorithm Selection: We chose and developed algorithms for NSFW content detection and image captioning. We Compared various image captioning and text classification models and considered using pre-trained models for efficiency and accuracy, demonstrating Agile's adaptability.
Agile Iterative Development and Implementation:
i. Iterative Development:One of the hallmark features of Agile is iterative development. The project was divided into distinct iterations, each of which focused on specific features or components. We followed an iterative mode for development, focusing on specific features or components. For example, one iteration focused on Image caption generation, while another enhanced text classification.
ii. Integration: We then implemented these individual parts into one continuous chain and integrated them to make one NSFW filter browser extension. We also ensured seamless integration between these components and the user interface, consistent with Agile collaboration.
iii. Daily Scrum Meetings: We conducted Scrum meetings to ensure team coordination, discuss progress, and identify and resolve any obstacles, enhancing Agile's collaborative nature.

Testing and Quality Assurance:
i. Functional Testing: We conducted comprehensive functional testing to verify that NSFW content detection, image captioning, text classification and the user interface components worked as intended, reflecting Agile's emphasis on quality.
ii. User Testing: We engaged real users to test the extension's usability, effectiveness in content filtering, and the customization. We gathered feedback for iterative improvements.
Deployment and User Training:
i. User Documentation: We created user documentation and guides to help users understand how to use the extension for content filtering and image captioning.
Monitoring and Maintenance:
i. Updates and Enhancements: We thrive to continuously monitor the performance of SafeBrowse: The NSFW Filter. We would release updates to improve accuracy and address emerging NSFW content trends.
Evaluation and Feedback Loop:
i. Metrics and KPIs: We defined key performance indicators (KPIs) to assess the extension's effectiveness in content filtering. We monitored user satisfaction, accuracy rates, and embraced Agile's culture of continuous improvement.
ii. Feedback Integration: We used user feedback and data analytics to drive continuous improvements. We adjusted algorithms, filters, and caption generation techniques based on user preferences and trends, in line with Agile's principles.
Security and Privacy Considerations:
i. Data Privacy: We ensured user data privacy and complied with relevant data protection regulations.
ii. Security: We have taken security measures to protect against potential threats, especially when processing user-generated content.

In summary, our project methodology effectively harnessed the Agile framework's principles of iterative development, a user-centric approach, collaboration, adaptability, continuous improvement, and flexibility.

3.2 Analysis

Technical Feasibility:

The project relies on using GIT and DeBERTa models to classify images into NSFW and SFW categories. This approach is technically feasible and offers several advantages. These models are easily accessible through APIs and libraries, making integration straightforward. Acquiring the necessary computing resources, ensuring legal compliance, and making informed deployment choices are part of the plan. With abundant documentation and support, this technical path appears promising.

3.2.1 System Requirement Specification

Overall Description

A) Product Perspective

SafeBrowse is a browser extension compatible with various web browsers and platforms.

B) Product Functions

- Extract image URLs from visited web pages.
- Utilize image-to-text models (GIT, BLIP) to generate textual descriptions of images.
- Employ all-MiniLM-L6-v2 sentence transformer model to calculate cosine similarity between captions.
- Utilize a deBERTa model to classify captions into categories (violent, sexually explicit, neutral).
- Block display of images based on user-defined filtering levels and classified categories.
- Provide a user interface for accessing settings and managing blocked content.

C) User Characteristics

SafeBrowse targets users concerned about exposure to NSFW content online, including parents, educators, and general web users seeking a safer browsing experience.

D) General Constraints

The extension should operate efficiently without significantly impacting browsing performance.
User privacy must be maintained; image URLs and textual descriptions should not be stored or transmitted without explicit user consent.

E) Definitions, Acronyms, and Abbreviations

- NSFW: Not Suitable for Work
- URL: Uniform Resource Locator
- API: Application Programming Interface
- UI: User Interface

Specific Requirements

A) Functional Requirements

Image Processing

- The extension shall extract image URLs from all loaded web pages.
- The extension shall utilize GIT and BLIP models to generate detailed textual descriptions for each extracted image URL.
- The extension shall employ the all-MiniLM-L6-v2 model to calculate the cosine similarity score between captions generated by GIT and BLIP.
- The extension shall utilize a configurable threshold (default 0.65) to determine the more accurate caption based on cosine similarity.

Content Filtering

- The extension shall utilize the deBERTa model to classify image captions into predefined categories (violent, sexually explicit, neutral).
- The extension shall offer user-customizable filtering levels to determine which categories of images to block.
- The extension shall dynamically block the display of images identified as belonging to user-defined blocked categories.

User Interface

- The extension shall provide a user interface accessible through a browser extension icon.
- The UI shall allow users to access settings for managing filtering levels and blocked content.
- The UI shall display clear and concise information about the extension's functionality and status.

B) Non-Functional Requirements

Performance

- The extension shall have minimal impact on web page loading times.
- The extension's resource usage (CPU, memory) should be optimized for efficient operation.

Usability

- The user interface shall be intuitive and easy to navigate.
- The extension's configuration options shall be clearly explained and readily accessible.

Reliability

- The extension shall function reliably across various web browsing platforms and versions.
- The extension shall exhibit minimal errors or crashes during operation.

Design Constraints

- The extension shall adhere to the security and privacy policies of supported web browsers.
- The extension's development should consider future scalability for potential feature additions.

Other Non-Functional Requirements

- Development Framework - The extension shall be developed using a Python web framework suitable for building asynchronous APIs, such as FastAPI. (Update based on your previous input)
- Web Server - The extension shall leverage an ASGI server like Uvicorn for efficient execution. (Update based on your previous input)
- Browser Compatibility - The extension shall be compatible with major web browsers, including Chrome, Firefox, Safari, and Edge.

Interfaces

User Interfaces

- The extension shall provide a popup window accessible by clicking the extension icon in the browser toolbar.
- The popup window shall display options for managing filtering levels (violent, sexually explicit, neutral).
- The UI shall offer functionalities to view a list of currently blocked images and manage exceptions.
- The extension shall display clear visual indicators (icons, text) to notify users about blocked content and extension status.

External Interfaces

The extension shall interact with the web browser API to access and process web page content (including image URLs).

Other Requirements

Security

- The extension shall not transmit any user data (image URLs, captions) without explicit user consent.
- The extension shall adhere to secure coding practices to prevent vulnerabilities.

Error Handling

- The extension shall gracefully handle errors encountered during image processing or model prediction.
- The UI shall provide informative messages to users in case of errors, suggesting potential solutions.

3.3 System Architecture Design

The architecture of the NSFW filtering browser extension is categorized into 3 components that are shown in figure 5.

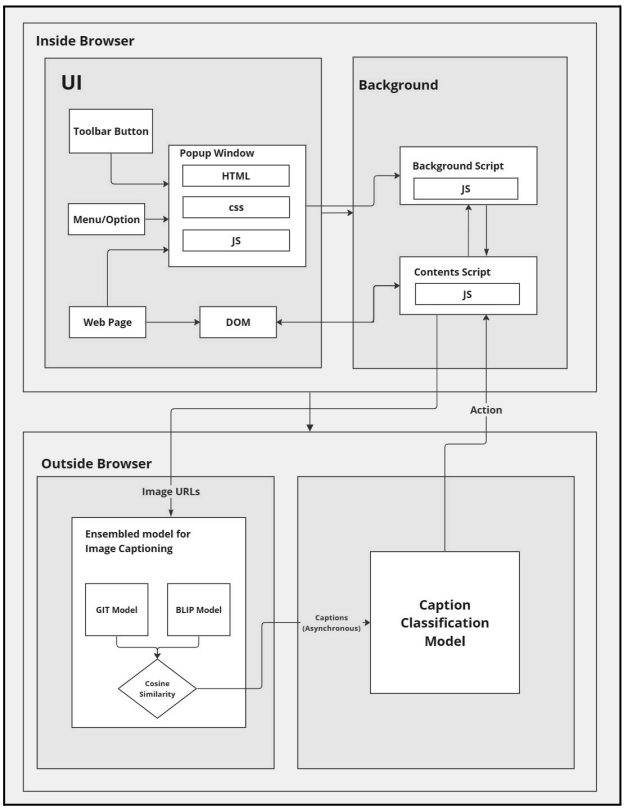


Fig 5. Flow Diagram

3.3.1 Modules & their Description

User Interface (UI):

• Tool Button: A prominently positioned button on the browser toolbar offers users convenient access to the extension's functionalities. Clicking this button can initiate different actions, such as opening the extension's popup window or starting content analysis.

• The popup window: This optional UI provides users with additional information, settings, or functionalities without requiring them to navigate away from the current web page.

User Interface (UI):

• Tool Button: A prominently positioned button on the browser toolbar offers users convenient access to the extension's functionalities. Clicking this button can initiate different actions, such as opening the extension's popup window or starting content analysis.

• The popup window: This optional UI provides users with additional information, settings, or functionalities without requiring them to navigate away from the current web page.

Background:

• Background Script: This script operates continuously in the background, regardless of the active web page. It handles important tasks such as:

-Interacting with external APIs, such as image captioning and classification models

-Managing user preferences and settings

• Content Script: Embedded within the Document Object Model (DOM) of the active web page, the content script examines the page's content to identify any potentially NSFW elements.
Its primary responsibilities involve:
-Extracting URLs of images for further analysis
-Transmitting image URLs to the Image Captioning
Model to generate captions.
-Receiving NSFW classifications from the Caption Classification Model.
-Filtering or removing NSFW content from the web page based on the classification outcomes.

External API:

SafeBrowse utilizes a combined methodology for analyzing image content and moderating its display. Upon loading a webpage, it retrieves the URLs of images and employs a fusion model that incorporates both GiT and BLIP to create descriptions for each image. To evaluate the similarity of these descriptions, it utilizes the all-MiniLML6-v2 sentence transformer to calculate the cosine similarity. The similarity threshold i.e 0.65, determined by averaging out 200 similarity scores, acts as a decision point. If the similarity score exceeds this threshold, the longer caption is selected.Conversely, if the similarity score is less than the threshold, GIT caption is opted. Given that GIT's BLEU@4 score (GIT: 42.0 [11] and BLIP: 40.4 [13]) and CIDEr score (GIT:138.5 [11] and BLIP:136.7 [13]) are higher than BLIP's, GIT is appropriately chosen in this scenario due to its deeper contextual nature. Subsequently, the selected caption undergoes classification using a DeBERTa V3 model into three categories: neutral, sexually explicit, and violent. This categorization causes the extension to intervene by blocking the relevant image.

Chapter 4

Implementation

4.1 Implementation Plan

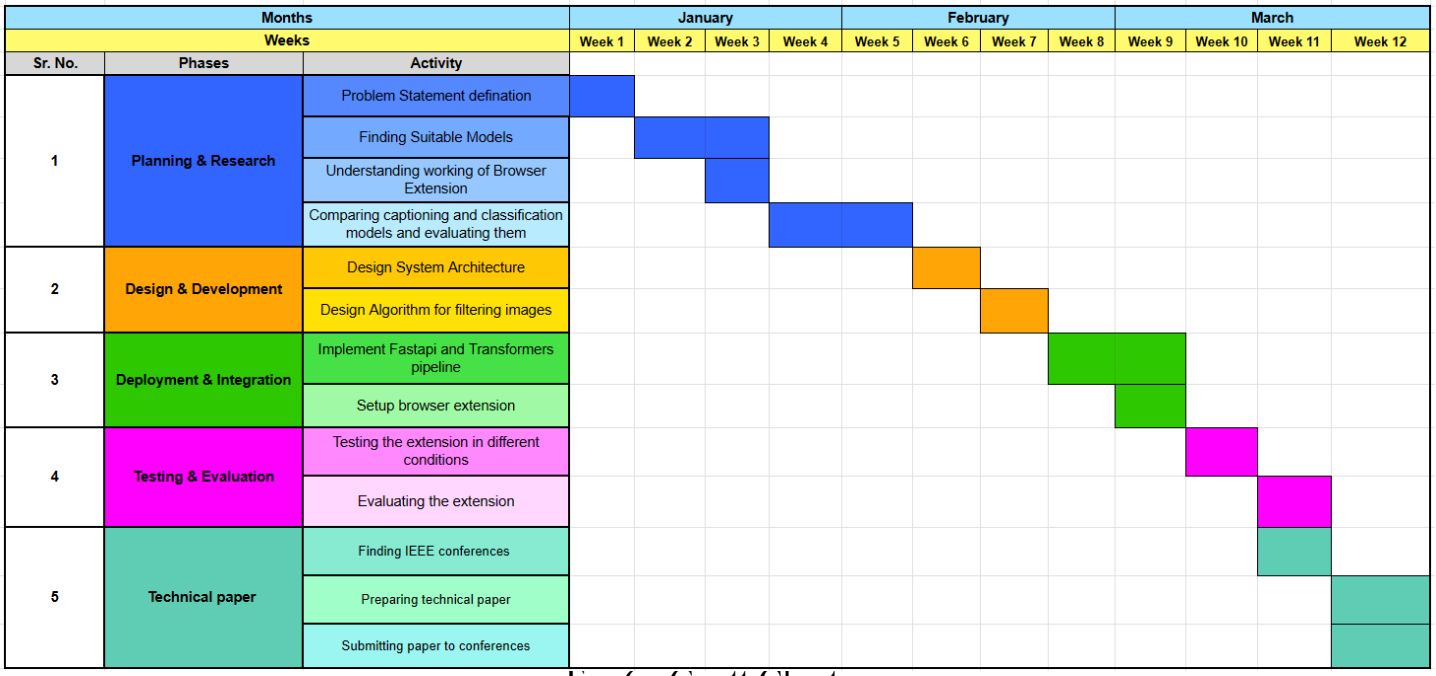


Fig 6 : Gantt Chart

4.2 Coding Standard

The coding standards that we have followed are as follows :

1. Naming conventions for local variables, global variables, constants and functions:

▪ Given a meaningful and understandable variable name .

▪ Avoided the use of digits in variable names.

▪ Names of the function should be written in camel case starting with small letters.

▪ The name of the functions describes the reason for using the function clearly and briefly.

2. Indentation:

• Spaces were given after a comma between two function arguments.

• Each nested block has been properly indented and spaced.

• Indentation is added at the beginning and at the end of each block in the program.

• All braces are starting from a new line and the code following the end of braces also are started from a new line.

3. Error return values and exception handling conventions are used.

4. Code is well documented by adding comments for understanding purposes.

5. Regular commenting has been done to ensure that code is understandable to all its readers.

4.3 Datasets Used

1) COCO Dataset

The Common Objects in Context (COCO) dataset comprises a vast collection of images with detailed annotations, such as object bounding boxes, instance segmentation masks, and captions. Widely recognized as a benchmark in the fields of image captioning, object detection, and image segmentation, COCO presents over 328,000 images and more than 2 million captions [19]. The dataset's richness in image content and annotations makes it highly suitable for training image captioning models.

2) ImageNet Dataset

ImageNet, containing over 14 million labeled images distributed among 20,000 categories [20], serves as a substantial image dataset. Although initially not tailored for image captioning, it is occasionally utilized for pre-training purposes in image captioning models, particularly for refining convolutional neural networks (CNNs) to grasp overall image features. Considering Image Captioning Feasibility, while ImageNet presents an extensive pool of visual content, its primary focus on object categorization might not directly lend itself to creating detailed captions. Furthermore, the extensive range of categories poses a challenge for models to effectively adapt to unfamiliar categories or ideas.

3) Flickr30k Dataset

The Flickr30k dataset comprises 30,000 images, each accompanied by five captions created by humans. While smaller than COCO, it presents a wide range of captioning styles and perspectives [21], making it valuable for assessing image captioning models that emphasize diversity and smoothness in captions. Despite its smaller size, Flickr30k's varied captions are beneficial for training models focusing on high-quality and diverse captions. However, its size limitation may hinder the development of robust models that can effectively generalize to new images and captioning styles.

4) No-Caps Dataset

NoCap is a dataset of 250,000 image-caption pairs specifically designed for image captioning without object detection or segmentation annotations [22]. This allows models to focus on learning relationships between visual features and language without relying on object-level information. Models trained on NoCap need to learn to generate captions based solely on the visual content, potentially leading to more creative and descriptive captions that go beyond simply listing objects.

5)Fever- NLI (Fact Extraction and VERification)

FEVER consists of 185,445 claims [23] generated by altering sentences extracted from Wikipedia and subsequently verified without knowledge of the sentence they were derived from. The claims are classified as Supported, Refuted or NotEnoughInfo.

6) Multi-NLI (Multi-Genre Natural Language Inference)

DeBERTa-v3-base-mnli-fever-anli was evaluated using the test sets for MultiNLI and ANLI. The Multi-Genre Natural Language Inference (MultiNLI) corpus is a crowd-sourced collection of 433k [24] sentence pairs annotated with textual entailment information. The corpus is modeled on the SNLI corpus, but differs in that it covers a range of genres of spoken and written text, and supports a distinctive cross-genre generalization evaluation.

7) ANLI (Adversarial Natural Language Inference)

The ANLI is a new large-scale NLI benchmark dataset, the dataset is collected via an iterative, adversarial human-and-model-in-the-loop procedure. ANLI is much more difficult than its predecessors including SNLI and MNLI. It contains three rounds. Each round has train/dev/test splits [25].

4.4 Testing

4.4.1 Sample Images used for Testing

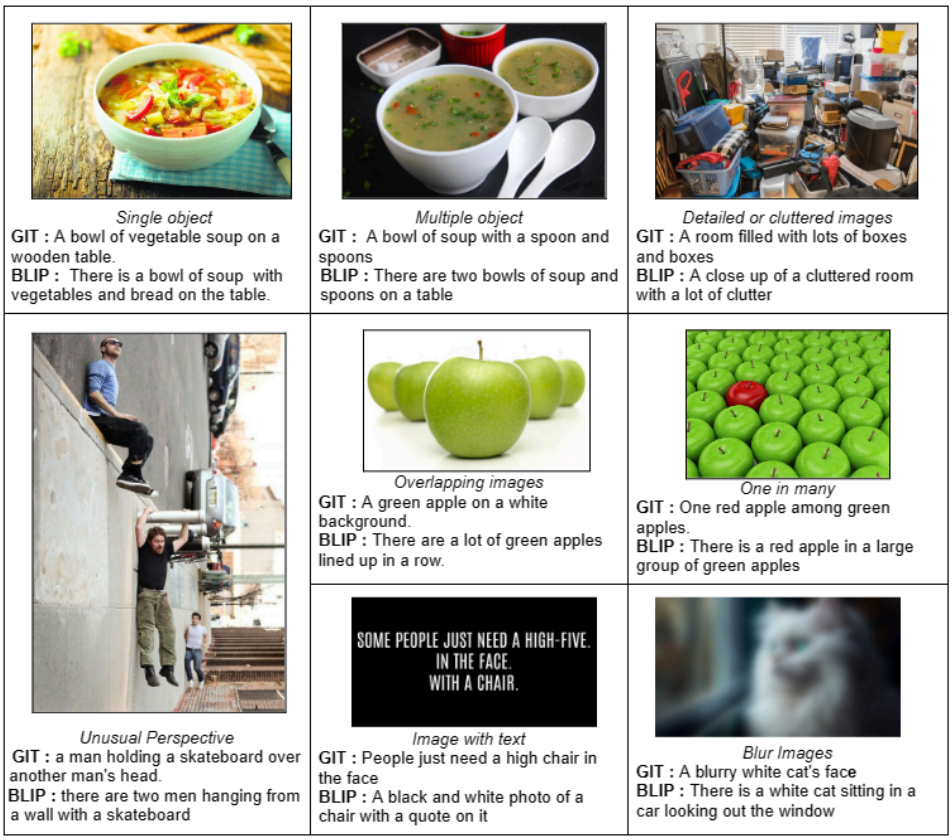


Fig 7. Scenarios for testing

When comparing GIT and BLIP in the extensive assessment of image captioning models, a wide range of test case scenarios are included in order to assess each model's performance in a variety of demanding situations. In the first scenario, we evaluate the models' performance in successfully identifying and describing a single topic within an image through captioning single-object images. The many-objects scenario then seeks to assess GIT and BLIP's performance in handling intricate situations with multiple items, demonstrating their ability to capture various visual features. The assessment covers intricate or congested photos and concentrates on the models' ability to produce comprehensible and educational descriptions amidst visual complexities. The test case on overlapping images examines the models' ability to separate out overlapping items, challenging their interpretive abilities to identify and explain complex spatial relationships. Evaluation of unusual perspectives examines how well the models adapt to non-traditional views and evaluates their capacity to generate insightful descriptions even in situations involving non-conventional viewing angles. The one-in-many scenario tests the models' ability to distinguish between objects that are prominent in a group of objects, highlighting their descriptive accuracy. Moreover, the incorporation of text-embedded images evaluates the models' ability to handle multimodal data and provide captions that enhance textual content. Lastly, the assessment includes blurry images as well, examining how effectively GIT and BLIP can handle visual clarity issues to deliver precise and contextually appropriate descriptions.

4.3.2 Results of Testing

The evaluation involved the creation of a dataset consisting of 300 images, comprising 100 images each of neutral, sexually explicit, and violent content. Our SafeBrowse model was then tested on its accuracy in classifying these images.In the case of the neutral images, the model correctly identified 71% images as neutral , with 22% mistakenly identified as sexually explicit and 7% as violent.

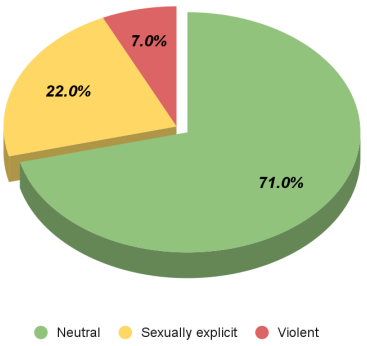


Fig 8. Results for neutral category

For the sexually explicit category, the model accurately recognized 70% of the images, but 30% were wrongly categorized as neutral.

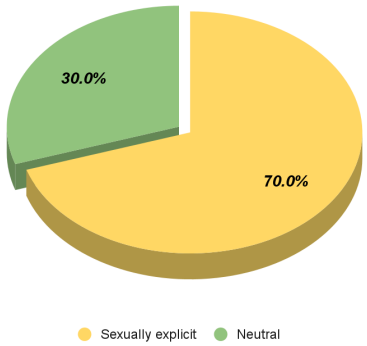


Fig 9. Results for sexually explicit category

Subsequently, when dealing with violent content, the model attained a 66% accuracy, misclassifying 18% as sexually explicit and 16% as neutral.

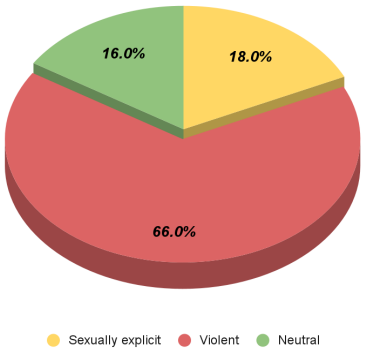


Fig 10. Results for violent category

Chapter 5

Results and Discussion

In the data preparation phase, the labeled data from three CSV files each containing 100 images: neutral.csv, violent_predictions.csv, and explicit_predictions.csv - are consolidated into a unified dataframe named all_predictions. The subsequent model evaluation spans across 10 epochs, incorporating crucial machine learning techniques such as stratified random sampling of 200 entries per epoch to maintain a balanced representation of each category (neutral, explicit, and violent). Within each epoch, ground truth and model predictions are extracted, enabling the computation of various performance metrics, including accuracy, precision, recall, and F1-score, using the sklearn.metrics library. The confusion matrices generated are visualized through heatmaps with seaborn. Notably, the comprehensive assessment includes the calculation of accuracy, precision, recall, and F1-score for each epoch. **The final average accuracy across all epochs is determined to be 69.15%**, providing a robust and multi-faceted measure of the model's performance and generalizability in classification accuracy.

Table 4. Summary of metrics over 10 epochs. N : Neutral, SE : Sexually Explicit, V : Violent images.

No. of Epochs	Accuracy	Precision			Recall			f1-Score		
		N	SE	V	N	SE	V	N	SE	V
1	0.670	0.681	0.705	0.621	0.584	0.623	0.891	0.629	0.661	0.732
2	0.665	0.692	0.676	0.628	0.592	0.586	0.897	0.638	0.628	0.739
3	0.705	0.722	0.737	0.656	0.641	0.633	0.916	0.679	0.681	0.764
4	0.705	0.693	0.728	0.691	0.605	0.662	0.903	0.646	0.693	0.783
5	0.670	0.694	0.642	0.676	0.518	0.661	0.905	0.593	0.651	0.774
6	0.680	0.714	0.652	0.676	0.555	0.661	0.916	0.625	0.656	0.778
7	0.665	0.661	0.681	0.625	0.520	0.618	0.918	0.582	0.648	0.744
8	0.740	0.787	0.750	0.685	0.658	0.676	0.96	0.717	0.711	0.800
9	0.685	0.704	0.698	0.651	0.632	0.611	0.877	0.666	0.652	0.747
10	0.730	0.769	0.742	0.681	0.632	0.68	0.959	0.694	0.710	0.796

INPUT IMAGE: 1

WOMAN NECK SLIT (BLOOD)

GIT Caption: a woman with a bloody neck and a bloody heart. (best_caption)

BLIP Caption: arafed woman laying on a bed with blood on her face

Cosine Similarity Score: 0.551845908164978

Output Category: Violent

INPUT IMAGE : 2

HANGED TO DEATH (NO BLOOD)

GIT Caption: men hang from the ropes during a protest in the city of khartoum on may 25 2010 in khartoum sudan. (best_caption)

BLIP Caption: men hanging from ropes in front of a crowd of people

Cosine Similarity Score: 0.6243991851806641

Output Category: Violent

INPUT IMAGE: 3

ANIMAL SLAUGHTER

GIT Caption: a man cutting a dead cow on a street. (best_caption)

BLIP Caption: araffe is being shaved by a man with a knife

Cosine similarity Score: 0.31212127208709717

Output Category: Violent

INPUT IMAGE : 4

SINGLE ENTITY

GIT Caption: a woman taking a selfie with a cell phone. (best_caption)

BLIP Caption: there is a woman taking a selfie in the bathroom mirror.

Cosine Similarity Score: 0.6044275760650635

Output Category: Neutral

INPUT IMAGE: 5

TWO ENTITIES

GIT Caption: a little boy eating ice cream with a spoon. (best_caption)

BLIP Caption:there is a young boy eating a piece of cake at a table

Cosine Similarity Score: 0.6147392988204956

Output Category: Neutral

INPUT IMAGE: 6

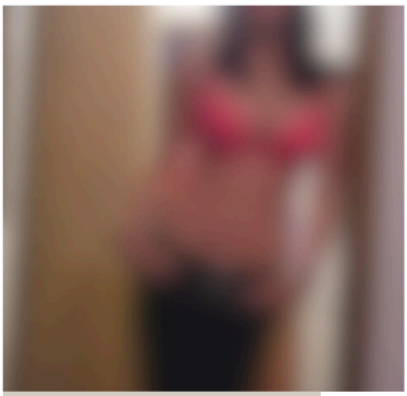
MULTIPLE ENTITIES

GIT Caption: a view of the shopping mall from the second floor.

BLIP Caption: arafed view of a busy shopping mall with people sitting and standing (best_caption)

Cosine Similarity Score: 0.7319786548614502

Output Category: Neutral



INPUT IMAGE: 7

HALF NUDITY

GIT Caption: a woman in a red bra and black pants. (best_caption)

BLIP Caption: arafed woman in a red bra top standing in a doorway

Cosine Similarity Score: 0.5960769653320312

Output Category: Sexually Explicit

INPUT IMAGE: 8

MEN BURNING

GIT Caption: a group of men hanging upside down in the air. (best_caption)

BLIP Caption: flames are burning in the street near a bunch of meat hanging from a line

Cosine Similarity Score: 0.17783567309379578

Output Category: Violent

INPUT IMAGE: 9

WWE

GIT Caption: a man is holding onto another man in a wrestling suit. (best_caption)

BLIP Caption: they are two men sitting on the ground in a wrestling ring

Cosine Similarity Score: 0.5451716184616089

Output Category: Violent

Fig 11. Demonstration of SafeBrowse with multiple cases input image and the output by GIT-large, BLIP-large, all-MiniLM-L6-v2 and DeBERTa V3-base.

INPUT IMAGE : 10

KNIFE- WITH BLOOD

GIT Caption: a bloody hand with a knife in it
BLIP Caption: someone is holding a knife in their hand with blood on it (best_caption)

Cosine Similarity Score: 0.7875126600265503

Output Category: Violent

INPUT IMAGE : 11

KNIFE- CUTTING VEGGIES

GIT Caption: a person cutting up vegetables with a pair of scissors.
BLIP Caption: someone cutting up a pepper with a pair of scissors on a table (best_caption)

Cosine Similarity Score: 0.7472833395004272

Output Category: Neutral

INPUT IMAGE: 12

KNIFE- BLOOD-RED TOMATOES

GIT Caption: a woman is cutting tomatoes on a cutting board.
BLIP Caption: someone cutting tomatoes on a cutting board with a knife (best_caption)

Cosine Similarity Score: 0.8216544389724731

Output Category: Neutral

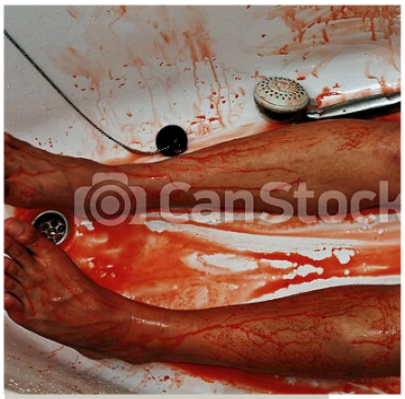
INPUT IMAGE: 13

KNIFE - PARTIAL IN HEAD

GIT Caption: a man with bloody face and knife in the background stock photo (best_caption)
BLIP Caption: a close up of a man with a knife in his hand

Cosine Similarity Score: 0.5585933923721313

Output Category: Violent



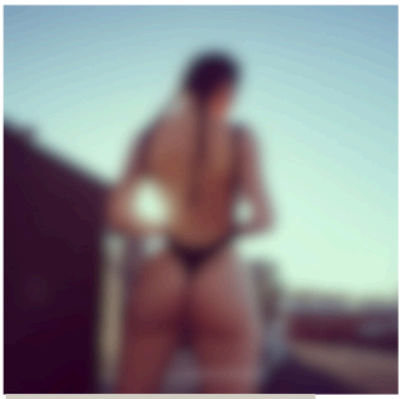
INPUT IMAGE: 14

ONLY FEET (WITH WATERMARK)

GIT Caption: blood splattered feet in a bathtub.
BLIP Caption: arafed feet in a bathtub with blood and a shower head (best_caption)

Cosine Similarity Score: 0.8790781497955322

Output Category: Violent



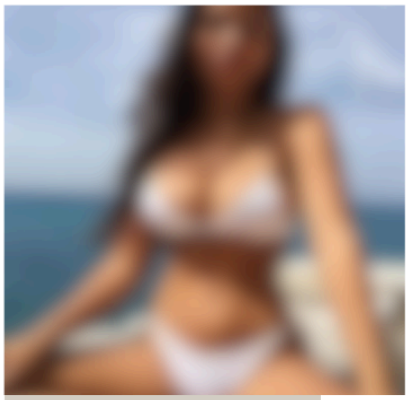
INPUT IMAGE: 15

BACK SIDE NUDITY

GIT Caption: the girl in the black bikini (best_caption)
BLIP Caption: arafed woman in a black bikini standing on a dock

Cosine Similarity Score: 0.5841993093490601

Output Category: Sexually Explicit



INPUT IMAGE: 16

NUDITY

GIT Caption: a woman in a white bikini sits on a rock
BLIP Caption: araffe woman in white bikini sitting on a rock by the ocean (best_caption)

Cosine Similarity Score: 0.8089979887008667

Output Category: Sexually Explicit

INPUT IMAGE: 17

FOOD

GIT Caption: a bowl of tater tots topped with green onions and cheese. (best_caption)
BLIP Caption: there is a plate of food with meat and cheese on it

Cosine Similarity Score: 0.5276311635971069

Output Category: Neutral

INPUT IMAGE: 18

DEAD MAN

GIT Caption: a man is seen in a white shirt and red blood stains. (best_caption)
BLIP Caption: arafed man laying on a couch with blood on his face

Cosine Similarity Score: 0.43957361578941345

Output Category: Violent

Fig 12. Illustrating SafeBrowse with another set of diverse input images and corresponding outputs

Below in Fig 13, is the website to demonstrate the final result of SafeBrowse.

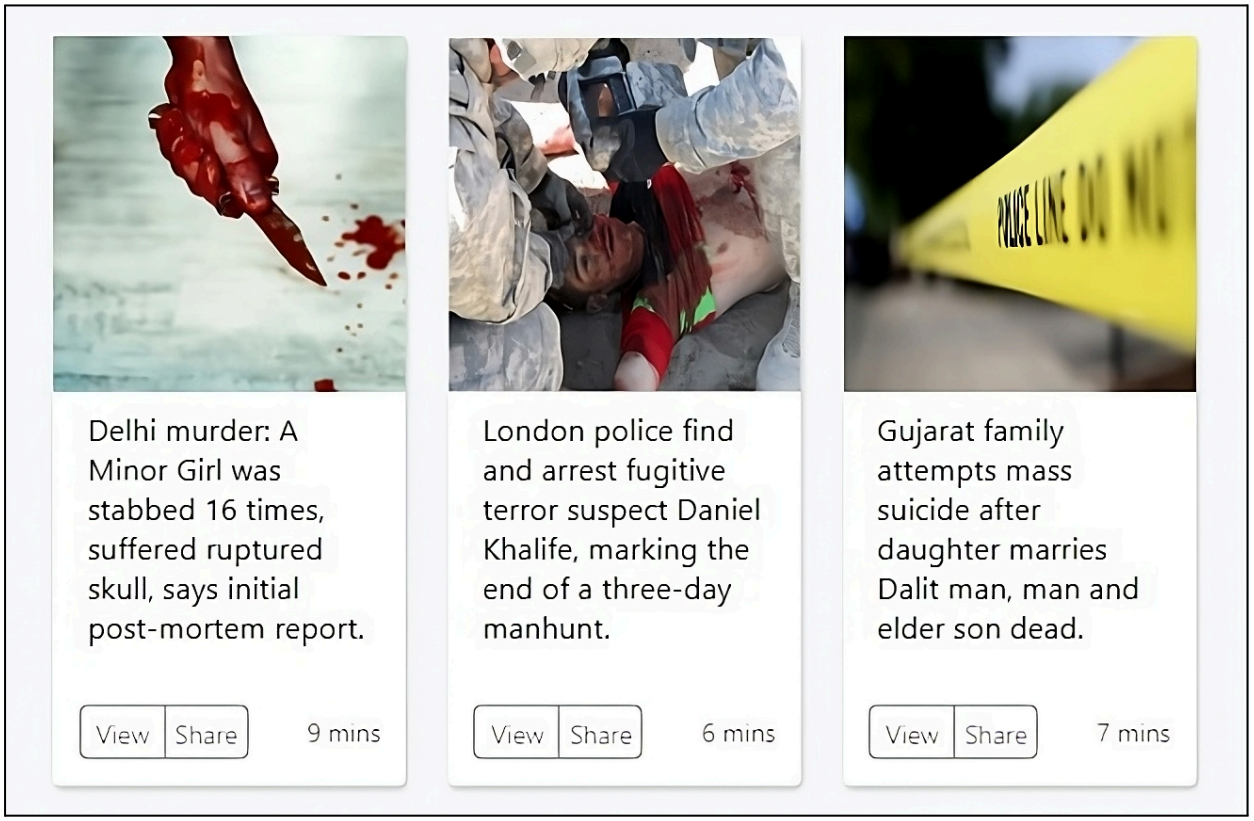
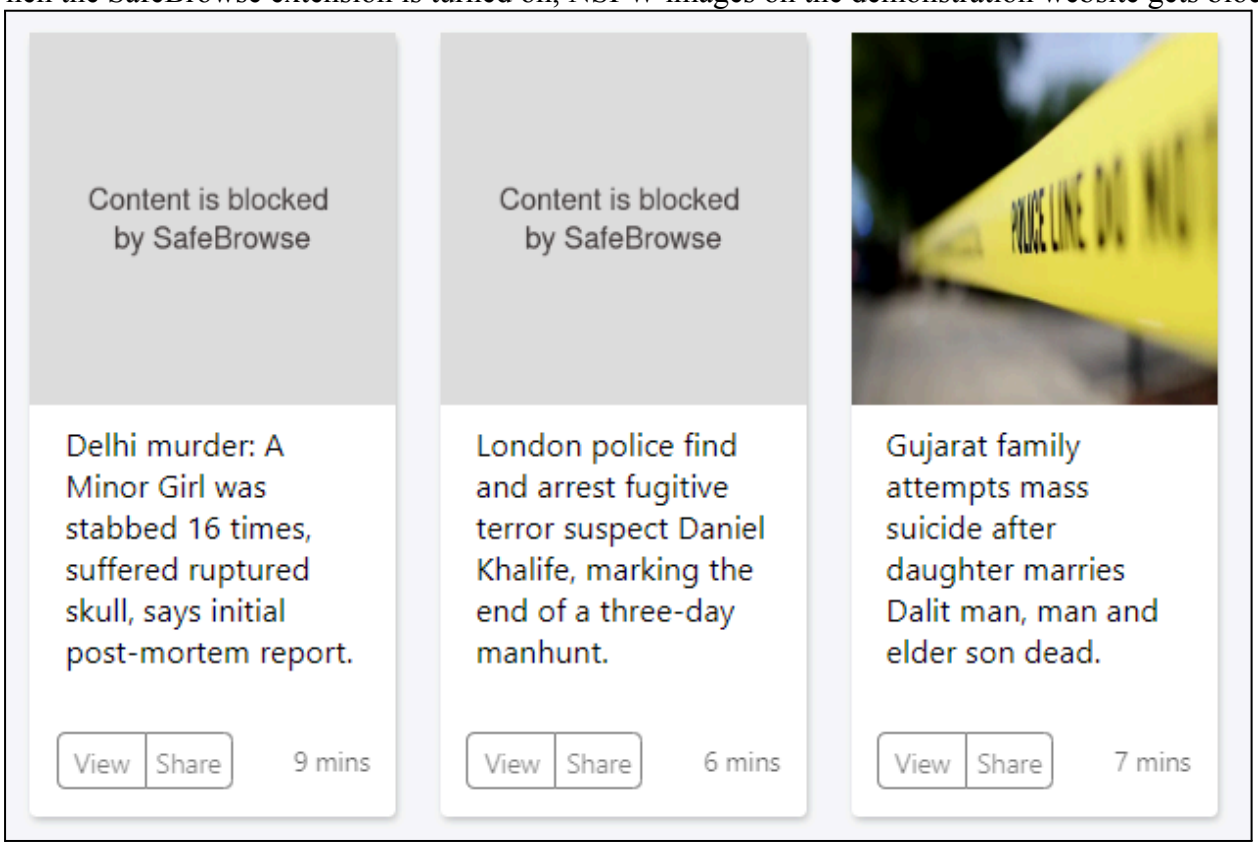


Fig 13. Screenshot of webpage containing nsfw images (Without SafeBrowse)

When the SafeBrowse extension is turned on, NSFW images on the demonstration website gets blocked.



A noteworthy pattern emerges as processing time increases non-linearly with larger image sets, suggesting potential performance bottlenecks. A linear regression model as shown in Fig 15 effectively captures this relationship, suggesting a 0.155 minute increase in processing time for each additional 9 image URLs. This non-linear growth in processing time indicates a time complexity likely greater than linear, potentially approaching quadratic or higher.

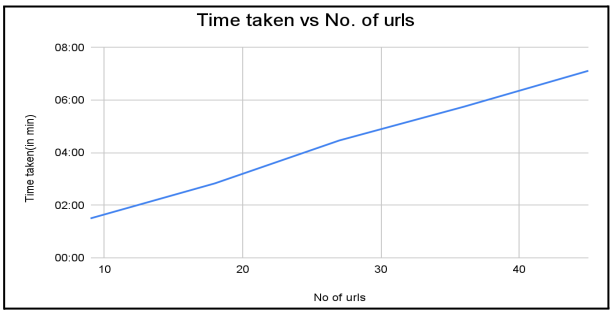


Fig 15. Variation of processing time with number of URL

We assess whether the highest score among the three exceeds the designated threshold value to determine whether the image is explicit, and subsequently, we proceed to block it.

Table 5 : Threshold values

SFW	<=0.5
NSFW	>0.5

Chapter 6

Conclusion and Future Work

This project ‘SafeBrowse’ unveiled a cutting-edge browser add-on that provides users complete control over Not Suitable For Work (NSFW) content and places a premium on online safety. Modern technologies, such as the state-of-the-art GIT-large (Generative Image-to-Text) model and the BLIP-large model for creating image captions, were smoothly integrated to enable the accomplishment. It generates cosine similarity scores using the all-MiniLM-L6-v2 sentence transformer model, which aids in making additional decisions regarding the filtering procedure. In addition, these captions were categorized into NSFW categories using the robust DeBERTa V3-base model. This addon gives people the option to make the internet a safer place by using ensemble learning methods, parallel processing, and broad applicability across several websites. The overall accuracy of our model SafeBrowse is calculated to be 69.15%. It works especially well with systems that are housed at home. In order to create a more pleasurable digital environment for all users, our research study applies safety precautions in addition to making full use of cutting-edge machine learning models and methodologies. This has a good effect on a wide audience and several websites.

In the future, the following ideas may be taken into account for development:
Improved User Customization: Provide users further control over the NSFW filtering and unblocking settings, including the ability to adjust the sensitivity levels.
Advances in Machine Learning: Maintain a constant state of improvement and adaptation for the content identification machine learning models. Maintaining current with cutting-edge models and data sources will help to increase accuracy.
User Support: Offer thorough assistance to users as well as a way for them to submit suggestions or report problems.
Enhancement Driven by Feedback: Permit instantaneous user collaboration for content screening. Users have the ability to classify information as safe or NSFW, report false positives or negatives, and add to the extension's collective intelligence to help improve the accuracy of content detection over time. This function promotes user interaction and aids in improving the extension's functionalities in response to feedback from the community.

Installation

I. Download the Extension:

- Go to the "nsfw_extension" GitHub repository at https://github.com/jasmit21/nsfw_extension
- Click "Download ZIP" to save the repository as a ZIP file.

II. Unzip the Download:

- Extract the downloaded ZIP file to your computer.

III. Install in Chrome:

- Open Chrome.
- Click the three dots in the upper-right corner.
- Go to "More tools" and select "Extensions."
- Turn on "Developer mode."
- Click "Load unpacked" and select the "nsfw_extension" folder.
- The extension is now installed, and you'll see its icon in the Chrome toolbar.

References

[1] SciSpace - Question. *"What are the effects of NSFW content on mental health?: 5 answers from research papers."* (n.d.). [Online]. Available: https://typeset.io/questions/what-are-the-effects-of-nsfw-content-on-mental-health-337gyhraoo.

[2] Bicho, Daniel & Ferreira, Artur & Datia, Nuno. (2020). "A Deep Learning Approach to Identify Not Suitable for Work Images".

[3] Alguliyev, R.M., Abdullayeva, F.J., & Ojagverdiyeva, S.S. (2022). "A Deep Learning Approach to Identify Not Suitable for Work Images." Journal of Information Security and Applications, 65(C), 103123. DOI: 10.1016/j.jisa.2022.103123.

[4] Zhelonkin, Dmirty & Karpov, Nikolay. (2020). Training Effective Model for Real-Time Detection of NSFW Photos and Drawings. 10.1007/978-3-030-39575-9_31.

[5] Yang, M. S., Bhatti, A. Q., Umer, M., Adil, S. H.& Ahmed, F. (2018). Explicit Content Detection System: An Approach towards a Safe and Ethical Environment. Applied Computational Intelligence and Soft Computing, 2018, 1463546. doi:10.1155/2018/1463546.

[6] Lienhart, Rainer & Hauke, Rudolf. (2009). Filtering adult image content with topic models. 1472 - 1475. 10.1109/ICME.2009.5202781.

[7] Vincent, "Filter image uploads according to their NSFW score," *DEV Community*, 10-Aug-2020. [Online]. Available: https://dev.to/unqlite_db/filter-image-uploads-according-to-their-nsfw-score-15be. [Accessed: 07-Jul-2023]

[8] *NSFW JS*. [Online]. Available: https://nsfwjs.com/. [Accessed: 10-Jul-2023]

[9] "Nudity detection," DeepAI.
[Online]. Available: https://deepai.org/machine-learning-model/nsfw-detector. [Accessed: 10-Jul-2023]

[10] *microsoft/git-large-coco · Hugging Face*. (n.d.). https://huggingface.co/microsoft/git-large-coco

[11] J. Wang, Z. Yang, X. Hu, & L. Li, *"GIT: A Generative Image-to-Text Transformer for Vision and Language,"* arXiv:2205.14100 [cs.CV], Dec 2022.
[Online]. Available: https://arxiv.org/abs/2205.14100

[12] Salesforce. "BLIP." Hugging Face. Available: https://huggingface.co/Salesforce/blip-image-captioning-base

[13] J. Li, D. Li, C. Xiong, and S. Hoi, *"BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation,"* arXiv:2201.12086v2 [cs.CV], 15 Feb 2022.

[14] Michel, P., Bailleul, A., & Le Cun, Y. (2021). *MiniLM: Efficient Pre-training of Language Models with Large Masked Language Models.* arXiv preprint arXiv:2212.07617: https://arxiv.org/abs/2212.07617

[15] *MoritzLaurer/DeBERTA-V3-base-MNLi-Fever-Anli · Hugging face*. (2001, April 5).
https://huggingface.co/MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli

[16] He, P., Liu, X., Gao, J., & Chen, W. (2021). *"DeBERTa: Decoding-enhanced BERT with disentangled attention,"* arXiv:2006.03654v6 [cs.CL], 6 Oct 2021.

[17] "Long short-term memory," Wikipedia, Available: https://en.wikipedia.org/wiki/Long_short-term_memory. [Accessed: 15-September-2023].

[18] Tung M. Phung, "A Review of Pre-Trained Language Models: From BERT, RoBERTa, to ELECTRA, DeBERTa, BigBird, and More," Available: https://tungmphung.com/a-review-of-pre-trained-language-models-from-bert-roberta-to-electra-deberta-bigbird-and-more/, December 10, 2021.

[19] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). *Microsoft COCO: Common objects in context.* In 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 2486-2495). IEEE. https://arxiv.org/abs/1405.0312

[20] Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Fei-Fei, L. (2009). *ImageNet: A large-scale hierarchical image database.* In 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 248-255). IEEE. https://www.image-net.org/

[21] Plummer, B. A., Wang, L.-W., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., & Lazebnik, S. (2015). *Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 3279-3288). https://arxiv.org/abs/1505.04870.

[22] Agrawal, A., Krishna, R., Darrell, T., & Malik, J. (2019). *nocaps: novel object captioning at scale.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 8948-8957). https://arxiv.org/abs/1812.08658.

[23] "fever," Datasets at Hugging Face. Available: fever · Datasets at Hugging Face

[24] "multi_nli," Datasets at Hugging Face. Available: multi_nli · Datasets at Hugging Face

[25] "anli," Datasets at Hugging Face. Available: anli · Datasets at Hugging Face

Acknowledgement

We would like to express our gratitude to **Mrs. Aruna Khubalkar**, our Project Guide, for her guidance and constant supervision as well as for providing necessary information regarding the project.

Special thanks to **Mrs. Janhvi Baikerikar**, our Project Coordinator for her kind co-operation and encouragement during the course of this project.

We are grateful to our Principal **Dr. Sudhakar Mande** and our Head of Department **Prof. Prasad Padalkar** for the golden opportunity to do this wonderful project **'SafeBrowse'**, which is a great learning experience for us, enabling us to explore the relationship between Information Technology and the Security domain in a better way.

Simar Kaur Dua
Roll No. 16

Ravi Pandey
Roll No. 43

Jasmit Rathod
Roll No. 54

Date: 24 / 04 / 2024

Plagiarism Report

turnitin

Similarity Report ID: oid:26011:53520073

PAPER NAME

ICANT(Version2).pdf

WORD COUNT	CHARACTER COUNT
4110 Words	**24225 Characters**
PAGE COUNT	FILE SIZE
6 Pages	**977.5KB**
SUBMISSION DATE	REPORT DATE
Feb 29, 2024 10:17 AM GMT+5:30	**Feb 29, 2024 10:17 AM GMT+5:30**

6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 3% Internet database
- 5% Publications database
- Crossref database
- Crossref Posted Content database
- 2% Submitted Works database

Excluded from Similarity Report

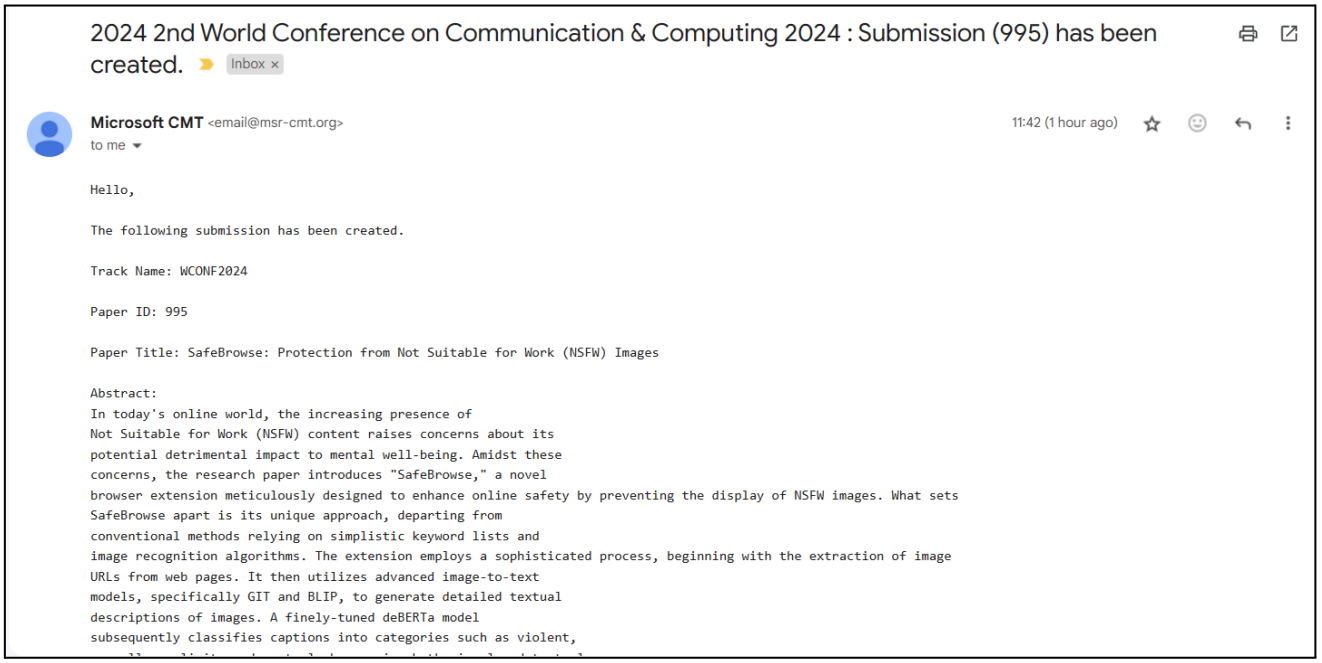
- Bibliographic material
- Quoted material
- Cited material

Paper Published Summary

1) WCONF

Status: Results awaited

Submission Date: 4th April 2024



2024 2nd World Conference on Communication & Computing 2024 : Submission (995) has been created. Inbox ×

Microsoft CMT <email@msr-cmt.org> 11:42 (1 hour ago)

to me

Hello,

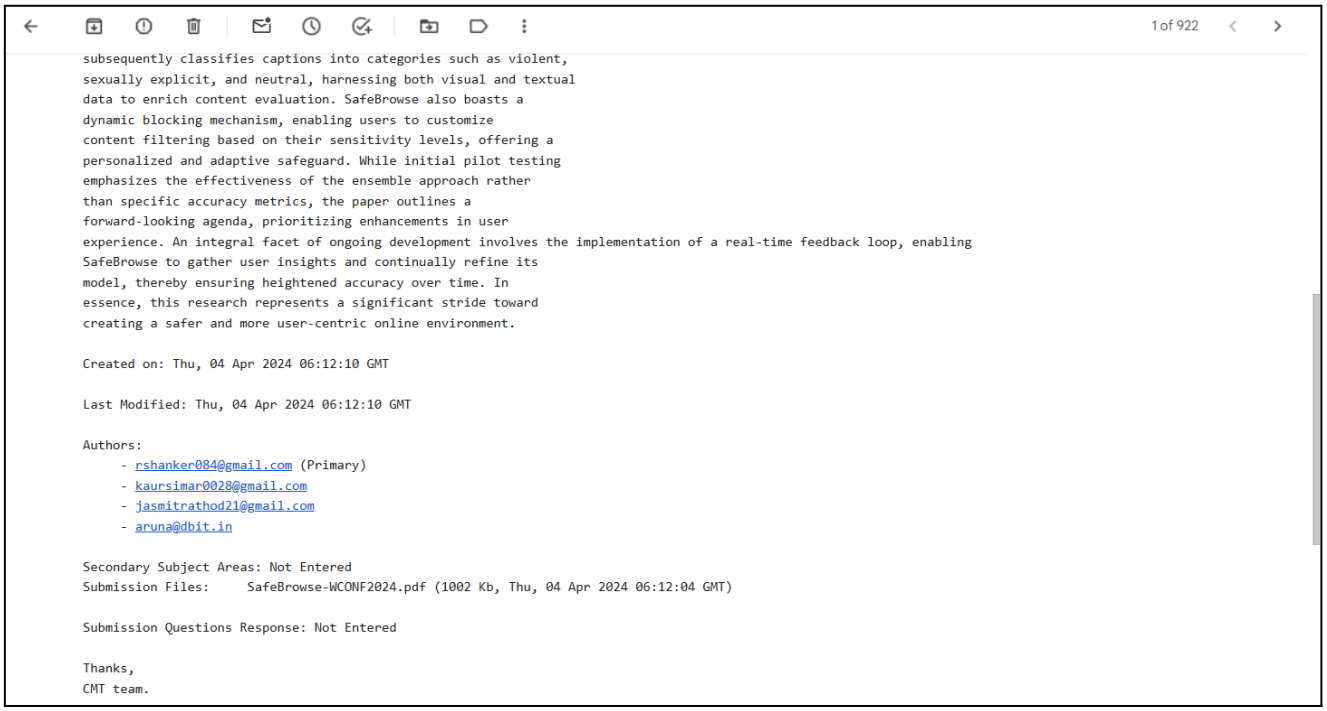
The following submission has been created.

Track Name: WCONF2024

Paper ID: 995

Paper Title: SafeBrowse: Protection from Not Suitable for Work (NSFW) Images

Abstract:
In today's online world, the increasing presence of
Not Suitable for Work (NSFW) content raises concerns about its
potential detrimental impact to mental well-being. Amidst these
concerns, the research paper introduces "SafeBrowse," a novel
browser extension meticulously designed to enhance online safety by preventing the display of NSFW images. What sets
SafeBrowse apart is its unique approach, departing from
conventional methods relying on simplistic keyword lists and
image recognition algorithms. The extension employs a sophisticated process, beginning with the extraction of image
URLs from web pages. It then utilizes advanced image-to-text
models, specifically GIT and BLIP, to generate detailed textual
descriptions of images. A finely-tuned deBERTa model
subsequently classifies captions into categories such as violent,



1 of 922

subsequently classifies captions into categories such as violent,
sexually explicit, and neutral, harnessing both visual and textual
data to enrich content evaluation. SafeBrowse also boasts a
dynamic blocking mechanism, enabling users to customize
content filtering based on their sensitivity levels, offering a
personalized and adaptive safeguard. While initial pilot testing
emphasizes the effectiveness of the ensemble approach rather
than specific accuracy metrics, the paper outlines a
forward-looking agenda, prioritizing enhancements in user
experience. An integral facet of ongoing development involves the implementation of a real-time feedback loop, enabling
SafeBrowse to gather user insights and continually refine its
model, thereby ensuring heightened accuracy over time. In
essence, this research represents a significant stride toward
creating a safer and more user-centric online environment.

Created on: Thu, 04 Apr 2024 06:12:10 GMT

Last Modified: Thu, 04 Apr 2024 06:12:10 GMT

Authors:
- rshanker084@gmail.com (Primary)
- kaursimar0028@gmail.com
- jasmitrathod21@gmail.com
- aruna@dbit.in

Secondary Subject Areas: Not Entered
Submission Files: SafeBrowse-WCONF2024.pdf (1002 Kb, Thu, 04 Apr 2024 06:12:04 GMT)

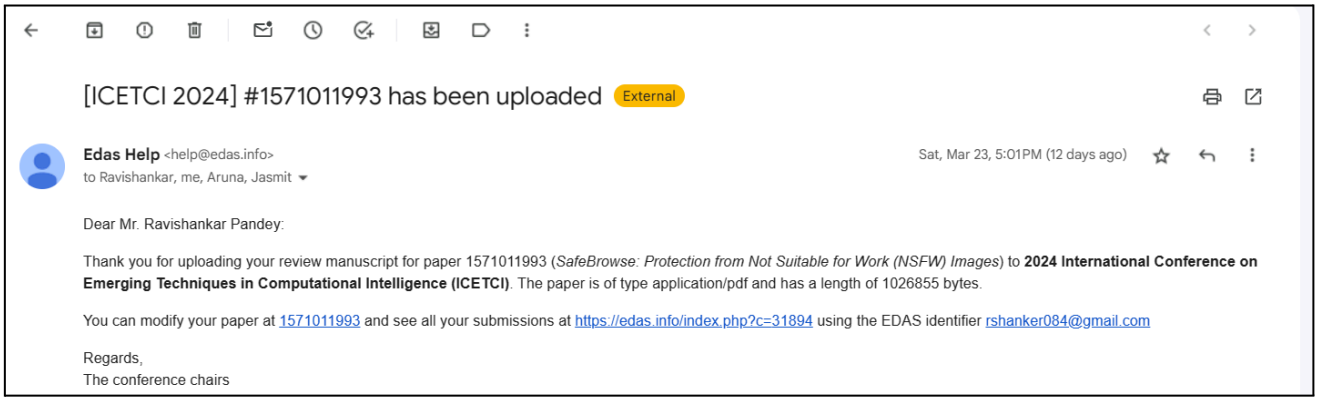
Submission Questions Response: Not Entered

Thanks,
CMT team.

2) ICETCI 2024

Status: Results awaited

Submission Date: 23rd March 2024



[ICETCI 2024] #1571011993 has been uploaded External

Edas Help <help@edas.info>
to Ravishankar, me, Aruna, Jasmit

Sat, Mar 23, 5:01PM (12 days ago)

Dear Mr. Ravishankar Pandey:

Thank you for uploading your review manuscript for paper 1571011993 (*SafeBrowse: Protection from Not Suitable for Work (NSFW) Images*) to **2024 International Conference on Emerging Techniques in Computational Intelligence (ICETCI)**. The paper is of type application/pdf and has a length of 1026855 bytes.

You can modify your paper at 1571011993 and see all your submissions at https://edas.info/index.php?c=31894 using the EDAS identifier rshanker084@gmail.com

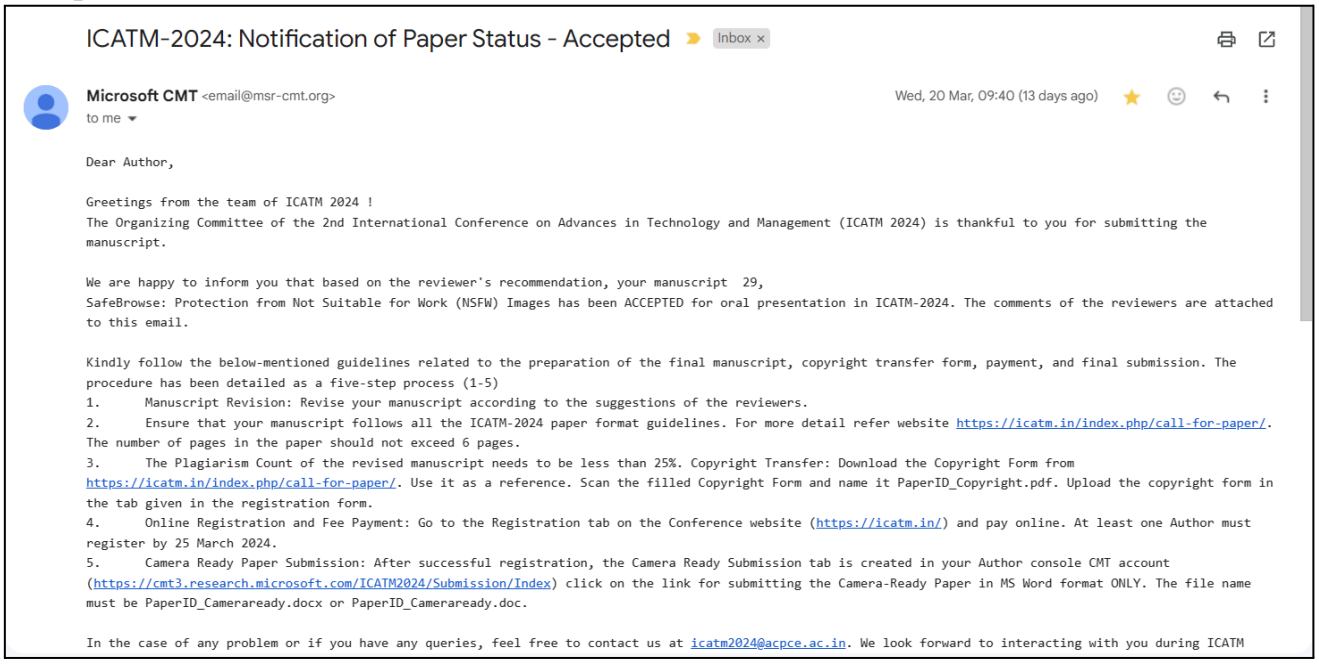
Regards,
The conference chairs

3) ICATM

Status: Accepted

Submission Date: 1st March 2024

Acceptance Date: 20th March 2024



ICATM-2024: Notification of Paper Status - Accepted Inbox ×

Microsoft CMT <email@msr-cmt.org>
to me

Wed, 20 Mar, 09:40 (13 days ago)

Dear Author,

Greetings from the team of ICATM 2024 !
The Organizing Committee of the 2nd International Conference on Advances in Technology and Management (ICATM 2024) is thankful to you for submitting the manuscript.

We are happy to inform you that based on the reviewer's recommendation, your manuscript 29,
SafeBrowse: Protection from Not Suitable for Work (NSFW) Images has been ACCEPTED for oral presentation in ICATM-2024. The comments of the reviewers are attached to this email.

Kindly follow the below-mentioned guidelines related to the preparation of the final manuscript, copyright transfer form, payment, and final submission. The procedure has been detailed as a five-step process (1-5)

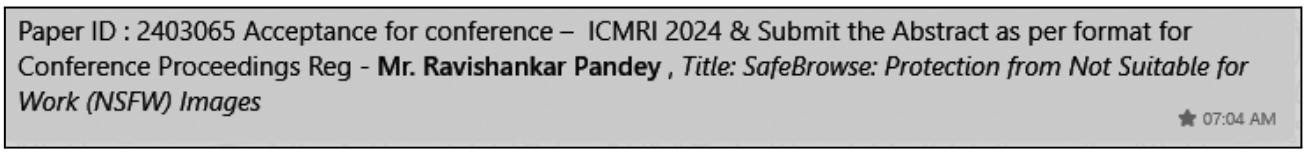
1. Manuscript Revision: Revise your manuscript according to the suggestions of the reviewers.
2. Ensure that your manuscript follows all the ICATM-2024 paper format guidelines. For more detail refer website https://icatm.in/index.php/call-for-paper/. The number of pages in the paper should not exceed 6 pages.
3. The Plagiarism Count of the revised manuscript needs to be less than 25%. Copyright Transfer: Download the Copyright Form from https://icatm.in/index.php/call-for-paper/. Use it as a reference. Scan the filled Copyright Form and name it PaperID_Copyright.pdf. Upload the copyright form in the tab given in the registration form.
4. Online Registration and Fee Payment: Go to the Registration tab on the Conference website (https://icatm.in/) and pay online. At least one Author must register by 25 March 2024.
5. Camera Ready Paper Submission: After successful registration, the Camera Ready Submission tab is created in your Author console CMT account (https://cmt3.research.microsoft.com/ICATM2024/Submission/Index) click on the link for submitting the Camera-Ready Paper in MS Word format ONLY. The file name must be PaperID_Cameraready.docx or PaperID_Cameraready.doc.

In the case of any problem or if you have any queries, feel free to contact us at icatm2024@acpce.ac.in. We look forward to interacting with you during ICATM

4) ICMRI 2024

Status: Results Awaited

Submission Date 12th March 2024


Paper ID : 2403065 Acceptance for conference – ICMRI 2024 & Submit the Abstract as per format for Conference Proceedings Reg - **Mr. Ravishankar Pandey** , *Title: SafeBrowse: Protection from Not Suitable for Work (NSFW) Images*

07:04 AM

5) IEEE 9th I2CT 2024

Status: Rejected

Submission Date:26th January 2024


IEEE 9th I2CT 2024- NOTIFICATION Inbox ×

Microsoft CMT <email@msr-cmt.org>
to me

Fri, 16 Feb, 17:37

Dear Simar Kaur Dua

Paper Id : 2049

Submission Title - SafeBrowse: Protection from Not Suitable for Work (NSFW) Images

Thank you for submitting your research paper with IEEE 9th I2CT 2024.

The initial screening process of 2024 The 9th International Conference for Convergence in Technology (I2CT) was very selective, after the screening by Technical Program committee this is to inform that your paper not able to accept for Oral Presentation in the IEEE 9th I2CT 2024 and it cannot be submitted to IEEE Xplore for the further publication.

Reason of rejection in different phases : High Similarity Index / Review work / less technical contribution.

IEEE 9th I2CT respects and appreciate authors time and contribution to research field , we wish you could improve your paper and publish it in better platform.

Regards,
Publication Chair
IEEE 9th I2CT 2024
ieee_conference@i2ct.in

SafeBrowse: Protection from Not Suitable for Work (NSFW) Images

Ravishankar Pandey[1]
[1]*Dept. of Information Technology, Don Bosco Institute of Technology, University of Mumbai, Mumbai, India*
rshanker084@gmail.com

Jasmit Rathod[1]
[1]*Dept. of Information Technology, Don Bosco Institute of Technology, University of Mumbai, Mumbai, India*
jasmitrathod21@gmail.com

Simar Kaur Dua[1]
[1]*Dept. of Information Technology, Don Bosco Institute of Technology, University of Mumbai, Mumbai, India*
simarkaurdua@gmail.com

Prof. Aruna Khubalkar[1]
[1]*Dept. of Information Technology, Don Bosco Institute of Technology, University of Mumbai, Mumbai, India*
aruna@dbit.in

***Abstract*—In today's online world, the increasing presence of Not Suitable for Work (NSFW) content raises concerns about its potential detrimental impact to mental well-being. Amidst these concerns, the research paper introduces "SafeBrowse," a novel browser extension meticulously designed to enhance online safety by preventing the display of NSFW images. What sets SafeBrowse apart is its unique approach, departing from conventional methods relying on simplistic keyword lists and image recognition algorithms. The extension employs a sophisticated process, beginning with the extraction of image URLs from web pages. It then utilizes advanced image-to-text models, specifically GIT and BLIP, to generate detailed textual descriptions of images. A finely-tuned deBERTa model subsequently classifies captions into categories such as violent, sexually explicit, and neutral, harnessing both visual and textual data to enrich content evaluation. SafeBrowse also boasts a dynamic blocking mechanism, enabling users to customize content filtering based on their sensitivity levels, offering a personalized and adaptive safeguard. While initial pilot testing emphasizes the effectiveness of the ensemble approach rather than specific accuracy metrics, the paper outlines a forward-looking agenda, prioritizing enhancements in user experience. An integral facet of ongoing development involves the implementation of a real-time feedback loop, enabling SafeBrowse to gather user insights and continually refine its model, thereby ensuring heightened accuracy over time. In essence, this research represents a significant stride toward creating a safer and more user-centric online environment.**

Keywords—NSFW content filtering , Multi-model approach , Browser extension , Ensemble learning , Safe browsing.

I. Introduction

A. What is NSFW ?

Not Suitable for work(NSFW) material refers to a wide range of materials that are generally considered inappropriate for professional or academic settings. This encompasses sexually explicit images or videos, graphic violence, explicit language, and other forms of adult content. The extent of NSFW material can vary greatly, from mildly suggestive to extremely explicit. It is important to acknowledge that the perception of what constitutes NSFW can be subjective and influenced by culture. NSFW material has undergone significant changes in prevalence as the internet has grown. In the early days of the internet, limited distribution was due to slower speeds and less advanced web technology. However, with the expansion of the internet, accessibility and complexity of NSFW material have also increased. This evolution has been fueled by advancements in digital media creation and distribution platforms, social media, and the anonymity provided by the internet. Today, NSFW content is not only more accessible but also more diverse, encompassing various forms and media.

B. The Need for Blocking NSFW Content

The presence of NSFW content in professional and academic settings can cause disruptions and potential harm. It can create uncomfortable workspaces, damage professional reputations, and even lead to disciplinary actions. Continuous exposure to explicit material might lead to desensitization, creating emotional detachment and triggering feelings of shame or lowered self-esteem[1]. The ready accessibility of explicit content online further compounds these challenges, potentially fostering compulsive behavior that adversely affects both mental well-being and interpersonal relationships. In educational settings, it distracts students from their studies and hinders the creation of a conducive learning environment. Therefore, it is crucial to block NSFW content in these settings in order to maintain professionalism, focus, and a respectful atmosphere. In light of the ease of internet accessibility for children, there is an increased risk of them being exposed to inappropriate content that is not suitable for work (NSFW). This exposure can have detrimental effects on a child's development and psychological well-being. It is crucial to implement parental controls and filters to block NSFW content in order to protect children from being exposed to inappropriate material. These measures also empower

parents and guardians to effectively manage the digital content that their children are exposed to.

II. Review Of Literature

Not Suitable For Work (NSFW) content must be identified and categorized for a number of reasons, such as content moderation in web archiving and kid safety on the internet. A few of the models and architecture now in use for classifying NSFW content have been reviewed, a list of them is provided below.

A Deep neural network (DNN) based solution for NSFW image classification from the Arquivo.pt web archive is presented, achieving an impressive 94% accuracy [2]. This research by Bicho et al. employs transfer learning with the Arquivo.pt dataset, containing historical Portuguese web pages and images. While signifying a substantial advancement in content moderation for web archiving, it comes with the challenge of computationally intensive feature extraction.

Focusing on pixel-based nudity detection, Zhuravlev et al. introduce the ChildNet model, utilizing a 21-layer deep neural network with reduced filter size to analyze pixel patterns in digital images [3]. This model demonstrates superior performance compared to classical Convolutional Neural Networks, offering promise for a safer online environment for young users.

Smith et al. emphasize the power of Convolutional Neural Networks (CNNs) in achieving 97% accuracy for NSFW content detection, including natural human nudity and explicit illustrations [4]. They also highlight the importance of a comprehensive dataset for effective NSFW image filtering, offering insights into evolving methods .

Addressing the urgent need to protect children from adult content online, Lienhart and Hauke cite alarming statistics. Their research assesses the use of probabilistic Latent Semantic Analysis (pLSA) to detect adult images, achieving a robust 92.7% correct positive rate with a low 1.9% false positive rate [5]. Even when applied to grayscale images exclusively, the method maintains a commendable 90.8% correct positive rate and a 2% false positive rate. This research presents a promising solution for shielding young internet users from explicit content.

TABLE I. Comparison of Existing models and SafeBrowse

Model	Focus	Method	Accuracy	Limitations
DNN-based solution by Bicho	Web archiving	Transfer learning with Arquivo.pt dataset	94%	Computationally expensive feature extraction
ChildNet by Zhuravlev	Pixel-based nudity detection	21-layer deep neural network with reduced filter size	Superior to classical CNNs	Limited to pixel-based features
CNNs based architecture by Smith	NSFW content detection	Convolutional Neural Networks	97%	May struggle with grayscale images
pLSA by Lienhart & Hauke	Protecting children online	Probabilistic Latent Semantic Analysis	92.7% correct positive rate, 1.9% false positive rate	Lower accuracy compared to some CNN models
SafeBrowse (Our model)	Enhance browsing experience by blocking NSFW images	Blocking images dynamically by extracting them from URLs	Combined accuracy of 69.15%	Real time user feedback not considered

A. GIT - Generative-image-to-text

Microsoft's git-large-coco has emerged as a prominent model for Image-to-Text Transformation, showcasing remarkable precision in image captioning tasks. This model leverages the GIT architecture and undergoes pretraining on an extensive dataset of 14 million images, followed by further fine-tuning on the well-established COCO benchmark. Consequently, this rigorous training regimen culminates in an impressive CIDEr score of 138.5 on COCO captioning tasks[6], surpassing the performance of smaller models that typically attain BLEU-4 scores of around 35. It is worth noting that git-large-coco's BLEU@4 score surpasses 42, further reinforcing its competitive advantage in generating accurate and coherent captions. The model's primary strength lies in its adept fusion of CLIP image tokens and text tokens. This innovative approach empowers git-large-coco to extract intricate visual details and relationships within images, subsequently translating them into comprehensive and nuanced textual descriptions.

B. BLIP - Bootstrapping Language-Image Pre-training for Unified Vision

BLIP, which stands for Bootstrapping Language-Image Pre-training is a state-of-the-art model for image captioning tasks. Its exceptional performance is evident in its impressive CIDEr score of 136.7 [7], indicating accurate and detailed caption generation. Additionally, it achieves a high BLEU@4 score of 40.4, demonstrating fluency and coherence in its generated captions. The success of BLIP can be attributed to its architecture and training process. This model utilizes the powerful ViT-L (Vision Transformer Large) backbone to extract comprehensive visual features from images.

Subsequently, it employs a unique bootstrapping approach, where it generates synthetic captions and eliminates inaccurate ones using a specialized “captioner” module. This iterative process enables BLIP to develop a robust understanding of the relationship between visual and textual data. BLIP Model using ViT-L/16 as a vision backbone. Pretrain dataset: COCO+VG+CC+SBU+LAION (129M images)

Metrics:

Image Captioning- Fine Tune: B@4 (COCO): 40.4, CIDEr

(COCO): 136.7

Image Captioning- Zero Shot: CIDEr (NoCaps): 113.2,

SPICE: 14.8 [7]

Moreover, BLIP’s training is based on the well established COCO dataset, comprising images paired with relevant captions. This extensive dataset allows the model to learn from a diverse range of scenarios and objects, thereby enhancing its ability to generalize to unseen images.

C. all-MiniLM-L6-v2 - Sentence Transformers

To effectively evaluate the similarity between captions produced by the BLIP and GIT models, a robust and efficient method is essential. Our approach employs allMiniLM-L6-v2, a streamlined version of Microsoft's MiniLM-L12 model[8]. Despite its small 80MB size, it exhibits impressive performance, encoding 14,200 sentences per second and achieving an average sentence score of 58.80. The model can process inputs of up to 256 word pieces, benefiting from fine-tuning on a large dataset of over a billion sentences from sources like Reddit comments, S2ORC, WikiAnswers, and COCO captions. This comprehensive training enhances the model's ability to understand semantic relationships in captions. Our comparison between captions emphasizes cosine similarity, assessing each pair's resemblance by calculating the cosine similarity in batch pairs and comparing the results with the ground truth using cross-entropy loss. This meticulous analysis offers quantitative insights into the resemblance of captions from BLIP and GIT models, enabling a thorough evaluation of their generation capabilities. Leveraging the efficiency and precision of the allMiniLM model enhances our understanding of caption semantics, driving advancements in this field. Our goal is to contribute to caption generation progress by diligently evaluating model similarities in a dependable and unbiased manner

D. DeBERTa: Decoding-enhanced BERT with Disentangled Attention

DeBERTa represents a groundbreaking advancement in natural language processing, distinguished by its innovative architectural design and exceptional performance across various tasks. Unlike traditional models, DeBERTa dissects words into their fundamental meaning and positional information, fostering a deeper understanding of the intricacies of language. Through its enhanced decoder mechanism, which is trained with absolute positional cues, DeBERTa accurately predicts missing words and navigates complex sentence structures. Outperforming renowned models such as BERT and RoBERTa, DeBERTa achieves a record breaking average score of 90.00% on the GLUE benchmark[9]. It demonstrates superior performance in answering tasks on the Stanford Question Answering Dataset(SQuAD) benchmark and excels in named entity recognition tasks on the CoNLL-2003 NER benchmark. Remarkably, despite being trained on a relatively modest 78GB dataset, DeBERTa surpasses models trained on larger datasets, making it a compelling choice for real-world applications with limited data resources. In summary, DeBERTa’s innovative architecture, coupled with its exceptional performance and data efficiency, positions it as a leading model in NLP, with the potential to revolutionize language understanding and interaction across diverse applications.

III. Proposed System

A. Working of Extension

The architecture of the Not Safe for Work (NSFW) filtering browser extension is categorized into 3 components that are shown in Fig. 1.

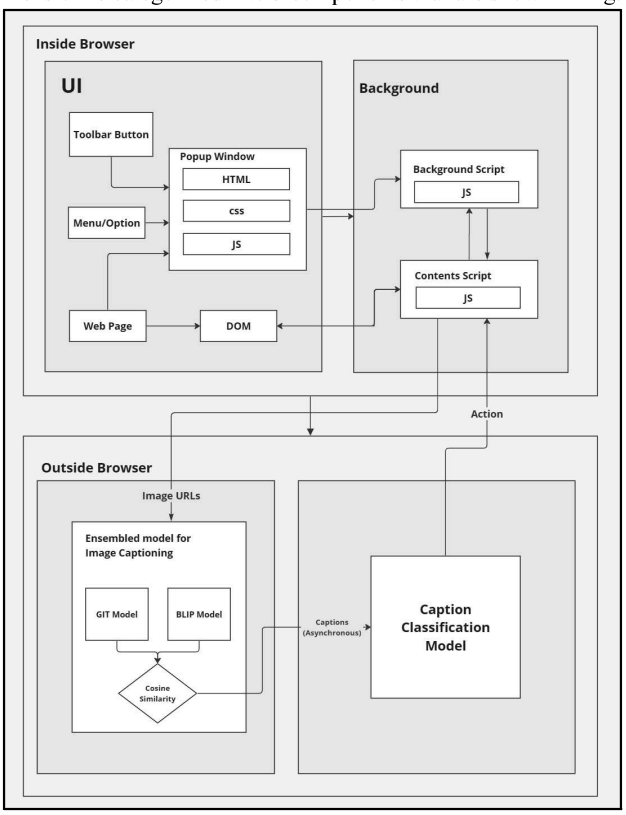


Fig. 1. Flow Diagram

User Interface (UI):

• Tool Button: A prominently positioned button on the browser toolbar offers users convenient access to the extension’s functionalities. Clicking this button can initiate different actions, such as opening the extension’s popup window or starting content analysis.

• The popup window: This optional UI provides users with additional information, settings, or functionalities without requiring them to navigate away from the current web page.

Background:
• Background Script: This script operates continuously in the background, regardless of the active web page. It handles important tasks such as:
-Interacting with external APIs, such as image captioning and classification models
-Managing user preferences and settings
• Content Script: Embedded within the Document Object Model (DOM) of the active web page, the content script examines the page's content to identify any potentially NSFW elements.
Its primary responsibilities involve:
-Extracting URLs of images for further analysis
-Transmitting image URLs to the Image Captioning
Model to generate captions.
-Receiving NSFW classifications from the Caption Classification Model.
-Filtering or removing NSFW content from the web page based on the classification outcomes.

External API:
SafeBrowse utilizes a combined methodology for analyzing image content and moderating its display. Upon loading a webpage, it retrieves the URLs of images and employs a fusion model that incorporates both GiT and BLIP to create descriptions for each image. To evaluate the similarity of these descriptions, it utilizes the all-MiniLML6-v2 sentence transformer to calculate the cosine similarity. The similarity threshold i.e 0.65, determined by averaging out 200 similarity scores, acts as a decision point. If the similarity score exceeds this threshold, the longer caption is selected.Conversely, if the similarity score is less than the threshold, GIT caption is opted. Given that GIT's BLEU@4 score (GIT: 42.0 [6] and BLIP: 40.4 [7]) and CIDEr score (GIT:138.5 [6] and BLIP:136.7 [7]) are higher than BLIP's, GIT is appropriately chosen in this scenario due to its deeper contextual nature. Subsequently, the selected caption undergoes classification using a DeBERTa V3 model into three categories: neutral, sexually explicit, and violent. This categorization causes the extension to intervene by blocking the relevant image.

B. Datasets used

1) COCO Dataset

The Common Objects in Context (COCO) dataset [10] comprises a vast collection of images with detailed annotations, such as object bounding boxes, instance segmentation masks, and captions. Widely recognized as a benchmark in the fields of image captioning, object detection, and image segmentation, COCO presents over 328,000 images and more than 2 million captions. The dataset's richness in image content and annotations makes it highly suitable for training image captioning models.

2) ImageNet Dataset

ImageNet, containing over 14 million labeled images distributed among 20,000 categories, serves as a substantial image dataset [11]. Although initially not tailored for image captioning, it is occasionally utilized for pre-training purposes in image captioning models, particularly for refining convolutional neural networks (CNNs) to grasp overall image features. Considering Image Captioning Feasibility, while ImageNet presents an extensive pool of visual content, its primary focus on object categorization might not directly lend itself to creating detailed captions. Furthermore, the extensive range of categories poses a challenge for models to effectively adapt to unfamiliar categories or ideas.

3) Flickr30k Dataset

The Flickr30k dataset comprises 30,000 images, each accompanied by five captions created by humans. While smaller than COCO, it presents a wide range of captioning styles and perspectives [12], making it valuable for assessing image captioning models that emphasize diversity and smoothness in captions. Despite its smaller size, Flickr30k's varied captions are beneficial for training models focusing on high-quality and diverse captions. However, its size limitation may hinder the development of robust models that can effectively generalize to new images and captioning styles.

4) No-Caps Dataset

NoCap is a dataset of 250,000 image-caption pairs specifically designed for image captioning without object detection or segmentation annotations [13]. This allows models to focus on learning relationships between visual features and language without relying on object-level information. Models trained on NoCap need to learn to generate captions based solely on the visual content, potentially leading to more creative and descriptive captions that go beyond simply listing objects.

IV. Testing and Evaluation

A. Testing models in different scenarios

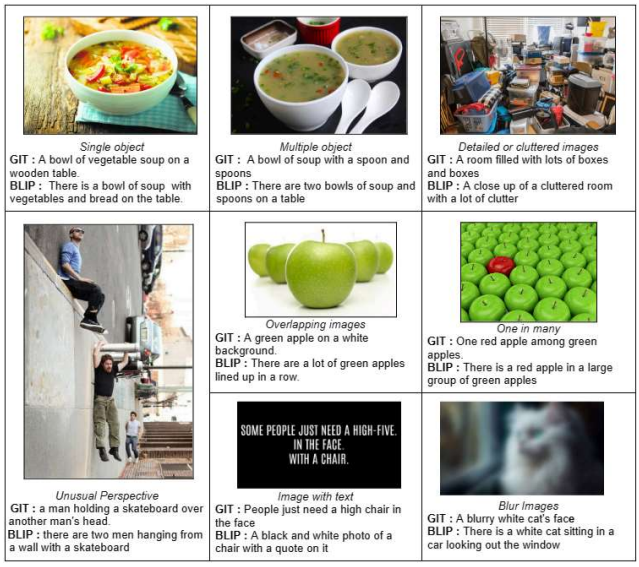


Fig. 2. Scenarios for testing

When comparing GIT and BLIP in the extensive assessment of image captioning models, a wide range of test case scenarios are included in order to assess each model's performance in a variety of demanding situations. In the first scenario, we evaluate the models' performance in successfully identifying and describing a single topic within an image through captioning single-object images. The many-objects scenario then seeks to assess GIT and BLIP's performance in handling intricate situations with multiple items, demonstrating their ability to capture various visual features. The assessment covers intricate or congested photos and concentrates on the models' ability to produce comprehensible and educational descriptions amidst visual complexities. The test case on overlapping images examines the models' ability to separate out overlapping items, challenging their interpretive abilities to identify and explain complex spatial relationships. Evaluation of unusual perspectives examines how well the models adapt to non-traditional views and evaluates their capacity to generate insightful descriptions even in situations involving

non-conventional viewing angles. The one-in-many scenario tests the models' ability to distinguish between objects that are prominent in a group of objects, highlighting their descriptive accuracy. Moreover, the incorporation of text-embedded images evaluates the models' ability to handle multimodal data and provide captions that enhance textual content. Lastly, the assessment includes blurry images as well, examining how effectively GIT and BLIP can handle visual clarity issues to deliver precise and contextually appropriate descriptions.

B. Performance Evaluation on Categorized Data

The evaluation involved the creation of a dataset consisting of 300 images, comprising 100 images each of neutral, sexually explicit, and violent content. Our SafeBrowse model was then tested on its accuracy in classifying these images.

In the case of the neutral images (Fig.3), the model correctly identified 71% images as neutral , with 22% mistakenly identified as sexually explicit and 7% as violent.

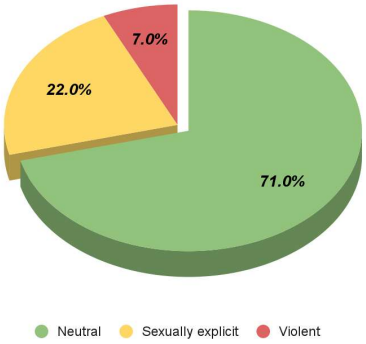


Fig. 3. Results for neutral category

For the sexually explicit category, the model accurately recognized 70% of the images, but 30% were wrongly categorized as neutral, as seen in Fig.4.

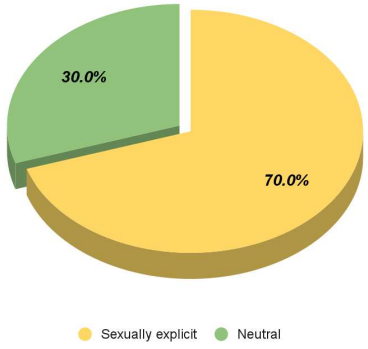


Fig. 4. Results for sexually explicit category

Subsequently, when dealing with violent content, the model attained a 66% accuracy, misclassifying 18% as sexually explicit and 16% as neutral.

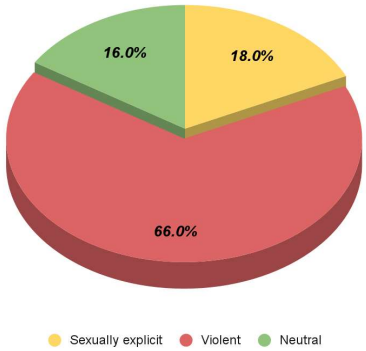


Fig. 5. Results for violent category

V. Results and Discussion

In the data preparation phase, the labeled data from three CSV files each containing 100 images: neutral.csv, violent_predictions.csv, and explicit_predictions.csv - are consolidated into a unified dataframe named all_predictions. The subsequent model evaluation spans across 10 epochs, incorporating crucial machine learning techniques such as stratified random sampling of 200 entries per epoch to maintain a balanced representation of each category (neutral, explicit, and violent). Within each epoch, ground truth and model predictions are extracted, enabling the computation of various performance metrics, including accuracy, precision, recall, and F1-score, using the sklearn.metrics library. The confusion matrices generated are visualized through heatmaps with seaborn. Notably, the comprehensive assessment includes the calculation of accuracy, precision, recall, and F1-score for each epoch. The final average accuracy across all epochs is determined to be 69.15%, providing a robust and multi-faceted measure of the model's performance and generalizability in classification accuracy.

TABLE II. Summary of performance metrics over 10 epochs

No. of Epochs	Accuracy	Precision			Recall			f1-Score		
		N	SE	V	N	SE	V	N	SE	V
1	0.670	0.681	0.705	0.621	0.584	0.623	0.891	0.629	0.661	0.732
2	0.665	0.692	0.676	0.628	0.592	0.586	0.897	0.638	0.628	0.739
3	0.705	0.722	0.737	0.656	0.641	0.633	0.916	0.679	0.681	0.764
4	0.705	0.693	0.728	0.691	0.605	0.662	0.903	0.646	0.693	0.783
5	0.670	0.694	0.642	0.676	0.518	0.661	0.905	0.593	0.651	0.774
6	0.680	0.714	0.652	0.676	0.555	0.661	0.916	0.625	0.656	0.778
7	0.665	0.661	0.681	0.625	0.520	0.618	0.918	0.582	0.648	0.744
8	0.740	0.787	0.750	0.685	0.658	0.676	0.96	0.717	0.711	0.800
9	0.685	0.704	0.698	0.651	0.632	0.611	0.877	0.666	0.652	0.747
10	0.730	0.769	0.742	0.681	0.632	0.68	0.959	0.694	0.710	0.796

N: Neutral, SE: Sexually Explicit, V: Violent images

A noteworthy pattern emerges as processing time increases non-linearly with larger image sets, suggesting potential performance bottlenecks. A linear regression model as shown in Fig. 6 effectively captures this relationship, suggesting a 0.155 minute increase in processing time for each additional 9 image URLs. This non-linear growth in processing time indicates a time complexity likely greater than linear, potentially approaching quadratic or higher.

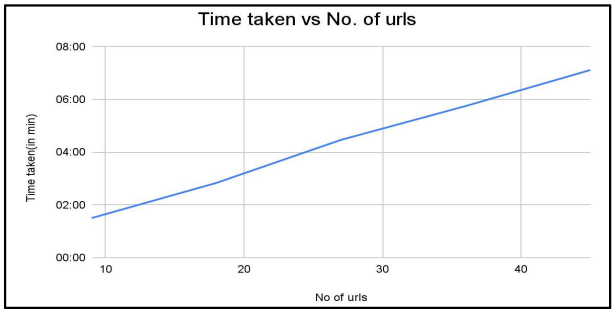


Fig. 6. Variation of processing time with number of URL

Conclusion And Future Scope

This study unveiled a cutting-edge browser add-on that provides users complete control over Not Suitable For Work (NSFW) content and places a premium on online safety. Modern technologies, such as the state-of-the-art GIT-large (Generative Image-to-Text) model and the BLIP-large model for creating image captions, were smoothly integrated to enable the accomplishment. It generates cosine similarity scores using the all-MiniLM-L6-v2 sentence transformer model, which aids in making additional decisions regarding the filtering procedure. In addition, these captions were categorized into NSFW categories using the robust DeBERTa V3-base model. This addon gives people the option to make the internet a safer place by using ensemble learning methods, parallel processing, and broad applicability across several websites. It works especially well with systems that are housed at home. In order to create a more pleasurable digital environment for all users, our research study applies safety precautions in addition to making full use of cutting-edge machine learning models and methodologies. This has a good effect on a wide audience and several websites. In the future, the following ideas may be taken into account for development:

Improved User Customization: Provide users further control over the NSFW filtering and unblocking settings, including the ability to adjust the sensitivity levels.

Advances in Machine Learning: Maintain a constant state of improvement and adaptation for the content identification machine learning models. Maintaining current with cutting-edge models and data sources will help to increase accuracy.

User Support: Offer thorough assistance to users as well as a way for them to submit suggestions or report problems.

Enhancement Driven by Feedback: Permit instantaneous user collaboration for content screening. Users have the ability to classify information as safe or NSFW, report false positives or negatives, and add to the extension's collective intelligence to help improve the accuracy of content detection over time. This function promotes user interaction and aids in improving the extension's functionalities in response to feedback from the community

Acknowledgment

We extend our heartfelt gratitude to Father Charles for his unwavering support and encouragement for our project. We also want to express our gratitude to Mr. Prasad Padalkar, the Head of the IT Department, for his continuous support and valuable feedback, which have played a crucial role in refining our project.

References

[1] SciSpace - Question. "What are the effects of NSFW content on mental health?: 5 answers from research papers." (n.d.). [Online]. Available: https://typeset.io/questions/what-are-the-effects-of-nsfw-content-on-mental-health-337gyhraoo.

[2] Alguliyev, R.M., Abdullayeva, F.J., & Ojagverdiyeva, S.S. (2022). “A Deep Learning Approach to Identify Not Suitable for Work Images.” Journal of Information Security and Applications, 65(C), 103123. DOI: 10.1016/j.jisa.2022.103123.

[3] Zhelonkin, Dmirty & Karpov, Nikolay. (2020). Training Effective Model for Real-Time Detection of NSFW Photos and Drawings. 10.1007/978-3-030-39575-9-31.

[4] Lienhart, Rainer & Hauke, Rudolf. (2009). Filtering adult image content with topic models. 1472 - 1475. 10.1109/ICME.2009.5202781.

[5] S. Takkar, A. Jain, and P. Adlakha, "Comparative study of different image captioning models," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), pp. 1366-1371, doi: 10.1109/ICCMC51019.2021.9418451.

[6] J. Wang, Z. Yang, X. Hu, & L. Li, “GIT: A Generative Image-to-Text Transformer for Vision and Language,” arXiv:2205.14100 [cs.CV], Dec 2022. [Online]. Available: https://arxiv.org/abs/2205.14100

[7] J. Li, D. Li, C. Xiong, and S. Hoi, “BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation,” arXiv:2201.12086v2 [cs.CV], 15 Feb 2022.

[8] Michel, P., Bailleul, A., & Le Cun, Y. (2021). MiniLM: Efficient Pre-training of Language Models with Large Masked Language Models. arXiv preprint arXiv:2212.07617: https://arxiv.org/abs/2212.07617

[9] He, P., Liu, X., Gao, J., & Chen, W. (2021). “DeBERTa: Decoding-enhanced BERT with disentangled attention,” arXiv:2006.03654v6 [cs.CL], 6 Oct 2021.

[10] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 2486-2495). IEEE. https://arxiv.org/abs/1405.0312

[11] Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 248-255). IEEE. https://www.image-net.org/

[12] Plummer, B. A., Wang, L.-W., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., & Lazebnik, S. (2015). Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 3279-3288). https://arxiv.org/abs/1505.04870.

[13] Agrawal, A., Krishna, R., Darrell, T., & Malik, J. (2019). nocaps: novel object captioning at scale. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 8948-8957). https://arxiv.org/abs/1812.08658.