IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.13511

Automated Harmful Content Control and Blocking System for Social Media

D. Tejaswi[1], K. Anusha[2], G. Vanaja[3], K. Deepa Sri Bhramaramba[4]

Assistant Professor, Department of Computer Science and Engineering,

Bapatla Women's Engineering College Bapatla, India[1]

Department of Computer Science and Engineering and Engineering,

Bapatla Women's Engineering College Bapatla, India[2-4]

Abstract: The rapid growth of multimedia content on digital platforms has created a need for efficient content moderation systems to prevent the spread of harmful material. This project presents an Automated Harmful Content Control and Blocking System that performs analysis of media before it is uploaded. The system allows users to upload images and videos through a web interface, where the content is processed using a Flask-based backend. For image analysis, OpenCV is used to perform preprocessing techniques such as grayscale conversion and pixel intensity evaluation. A threshold-based method is applied to determine whether the uploaded image contains potentially harmful content. If harmful content is detected, the system blocks the upload and notifies the user; otherwise, the file is stored successfully. The system also supports video uploads and provides a deletion feature for managing uploaded files. This approach ensures real-time moderation, reduces dependency on manual monitoring, and enhances platform safety. Although the current implementation uses basic image processing techniques, it can be extended with advanced machine learning models for improved accuracy in future developments.

Keywords: Harmful Content Detection,Image Processing, OpenCV,Flask Content Moderation,Social Media Safety

I. INTRODUCTION

The rapid growth of social media platforms has revolutionized the way people communicate, share information, and express opinions. Millions of users actively upload text, images, and videos every day, making social media an essential part of modern digital life. However, this widespread usage has also led to a significant increase in the circulation of harmful content such as abusive language, hate speech, nudity, misinformation, and other inappropriate materials. The uncontrolled spread of such content poses serious challenges to user safety, platform credibility, and societal well-being.

Traditional content moderation systems primarily rely on manual review or post-upload detection mechanisms. In theseapproaches, harmful content is identified only after it has already been published, which allows it to spread rapidly among users before any action is taken. This delay not only reduces the effectiveness of moderation but also exposes users to potentially harmful material. Furthermore, most existing systems focus on analyzing individual posts without considering repeated harmful behavior by the same user. As a result, users who frequently post inappropriate content are not effectively monitored or restricted.

Recent advancements in artificial intelligence and machine learning have introduced automated content detection techniques, including multimodal transformers, neural networks, and graph-based analysis. While these methods improve detection accuracy, they often require high computational resources and are not suitable for real-time implementation in lightweight systems. Additionally, many existing solutions are limited to specific content types, such as text or memes, and fail to provide a comprehensive approach that supports images, videos, and user-level moderation simultaneously.

To address these limitations, this project proposes an **Automated Harmful Content Control and Blocking System** that focuses on real-time moderation. Unlike conventional systems, the proposed approach analyzes content before it is uploaded, ensuring that harmful material is blocked at the source itself. The system allows users to upload images and videos through a web-based interface, where the content is immediately processed and evaluated. By integrating image processing techniques, the system identifies harmful visual patterns and prevents such content from being stored or displayed.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.13511

In addition to content-level moderation, the system introduces user-level control by tracking repeated harmful behavior. Based on predefined thresholds, the system can issue warnings, temporarily restrict accounts, or permanently block users who consistently violate platform guidelines. This dual-level moderation approach enhances both content filtering and user Accountability, Solution content moderation. By combining pre-upload detection with user behavior tracking, the system significantly improves the effectiveness of harmful content control and contributes to creating a safer and more trustworthy digital environment.

II. LITERATURE SURVEY

The problem of harmful content detection and moderation has gained significant attention in recent years due to the rapid growth of social media platforms. Researchers have proposed various techniques ranging from traditional machine learning methods to advanced deep learning and multimodal systems. This section provides a detailed analysis of existing approaches and their limitations.

CLARITY et al. (2026) proposed a lightweight cross-modal transformer model for harmful content detection. The primary purpose of this work was to improve detection accuracy by combining multiple data modalities such as text, images, and audio. By integrating these modalities, the system was able to understand contextual relationships more effectively than single-modality approaches. However, the model mainly focuses on classification after content is generated and does not provide a mechanism for preventing harmful content before it is uploaded. Additionally, it lacks user-level moderation features.

Li et al. (2025) developed a dual-branch neural network aimed at detecting harmful memes. The purpose of this research was to analyze both textual and visual components of memes simultaneously, as harmful intent often arises from the combination of these elements. The system processes image features and embedded text separately and then merges them for final classification. Although effective for meme detection, this approach is limited to a specific type of content and does not support general image or video moderation. It also does not address real-time blocking of harmful content.

Kumar et al. (2025) introduced the MAHGA (Multi-Aspect Heterogeneous Graph Analysis) model for harmful speech detection. The purpose of this work was to analyze relationships between users, posts, and interactions using graph-based techniques to identify patterns of harmful behavior. This model provides deeper insights into user activity and network interactions. However, due to its high computational complexity, it is not suitable for real-time applications. Moreover, it focuses primarily on textual and interaction-based data rather than multimedia content.

Smith et al. (2024) proposed a supervised machine learning approach for detecting sensitive and harmful textual content on social media platforms. The purpose of this system was to classify user-generated text into harmful and non-harmful categories using labeled datasets. While this approach is efficient for text moderation, it does not support image or video content. Additionally, it performs detection only after content has been posted, which limits its ability to prevent the spread of harmful material.

Zhang et al. (2024) developed MTikGuard, a transformer-based multimodal system designed for short video content moderation. The purpose of this research was to analyze video frames, audio signals, and embedded text using advanced techniques such as optical character recognition (OCR). This comprehensive approach improves detection accuracy across multiple content types. However, the system is computationally intensive and requires high processing power, making it difficult to implement in real-time environments. It also lacks mechanisms for tracking repeated harmful behavior by users.

Goodfellow et al. (2016) introduced the concept of adversarial examples in machine learning. The purpose of this research was to demonstrate how small, imperceptible changes to input data can mislead machine learning models. This work highlights a major challenge in harmful content detection systems, as attackers can manipulate content to bypass detection. Although this research provides valuable insights, it does not directly address practical moderation or prevention mechanisms.

He et al. (2016) proposed the ResNet (Residual Network) architecture, which is widely used for image recognition tasks. The purpose of this work was to improve deep learning performance by enabling the training of very deep neural networks. While ResNet has been applied in harmful image detection, its high computational requirements make it less suitable for lightweight systems or real-time applications.

Devlin et al. (2019) introduced BERT (Bidirectional Encoder Representations from Transformers), a powerful model for natural language processing tasks. The purpose of this research was to improve contextual understanding of text by

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.13511

analyzing bidirectional relationships. BERT has been used in harmful text detection, but it does not address visual content and requires significant computational resources.

Tan and Le (2019) proposed EfficientNet, a scalable convolutional neural network model designed for efficient image classification. The purpose of this work was to achieve high accuracy with fewer parameters compared to traditional models. Although EfficientNet improves efficiency, it still requires training and computational resources, which may not be practical for simple real-time systems.

Despite the advancements made by these researchers, most existing systems focus on detecting harmful content after it has already been uploaded. Their primary purpose is classification rather than prevention. Additionally, many approaches rely on complex models that require high computational resources, making them unsuitable for real-time and lightweight applications. Most systems also lack integration between content detection and enforcement mechanisms such as blocking or deletion.

To address these limitations, the proposed system focuses on real-time content moderation by analyzing and filtering content before it is uploaded. The purpose of this system is to provide a simple, efficient, and practical solution that can be implemented using lightweight image processing techniques. By blocking harmful content at the source and providing user-level control features such as deletion, the system ensures improved safety and usability compared to existing approaches.

Human-in-the-loop systems have also been proposed, where automated detection is combined with human review for final decision-making. While this improves accuracy, it increases operational cost and reduces scalability. Such systems are not suitable for applications requiring fully automated and real-time moderation.

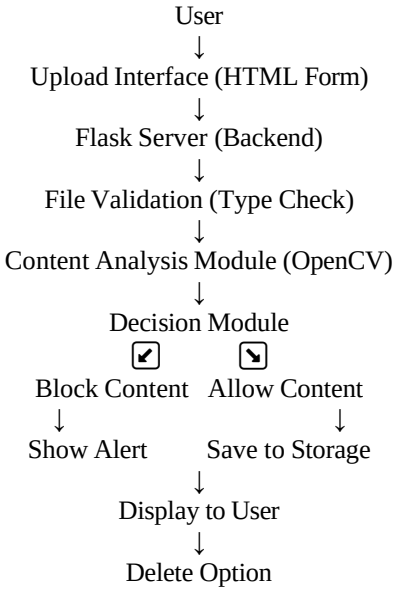
In conclusion, despite the wide range of techniques developed for harmful content detection, several challenges remain unresolved. These include high computational requirements, lack of real-time processing, limited support for multiple content types, and absence of effective enforcement mechanisms. The proposed system addresses these issues by providing a simple, efficient, and real-time solution that focuses on pre-upload content blocking. By using lightweight image processingTechniques and a user-friendly interface,the system ensures practical implementation while maintain effective content control.

III. PROPOSED SYSTEM ARCHITECTURE

The proposed system, titled **"Automated Harmful Content Control and Blocking System,"** is designed to detect and prevent harmful content before it is uploaded to the platform. Unlike existing systems that perform post-upload analysis, this system adopts a **proactive approach** by analyzing media in real time and blocking inappropriate content at the source.

The system allows users to upload images and videos through a web interface. The uploaded files are processed by a Flask-based backend, where image processing techniques using OpenCV are applied to detect harmful content. If harmful content is identified, the system immediately blocks the upload and notifies the user. Otherwise, the file is stored and displayed. Additionally, users are provided with a delete option to manage their uploaded content.

A. System Architecture



IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.13511

B. User

The user is the starting point of the system. The user interacts with the application through a web browser and performs actions such as selecting and uploading files. The system is designed to be user-friendly so that even non-technical users can easily use it. The user can upload images or videos and also view previously uploaded files. Additionally, the user has the ability to delete unwanted content, giving them full control over their data.

C. Upload Interface(HTMAL Format)

The upload interface is developed using HTML and acts as a bridge between the user and the backend system. It contains a file input field that allows users to select images or videos from their device. The form uses the POST method along with multipart encoding to send files securely to the server. This module ensures that the data is correctly packaged and transmitted. It also provides a simple and responsive interface for better user experience.

D. Flask Server(Backend Processing)

The Flask server is the core backend component that handles all requests from the user interface. When a user uploads a file, the request is sent to the Flask server, which processes the data and performs necessary operations. It manages routing, handles file uploads, performs validation, and communicates with other modules. Flask is lightweight and efficient, making it suitable for real-time applications. It also enables easy integration with image processing libraries like OpenCV.

E. File Validation Module

The file validation module ensures that only supported file types are processed by the system. It checks whether a file is selected and verifies its extension against a predefined list of allowed formats such as .jpg, .png, .jpeg, .mp4, .avi, and .mov. This prevents unauthorized or unsupported files from entering the system. Validation also improves system security by reducing the risk of malicious file uploads.

F. Content Analysis Module(OpenCV)

This is the most important module of the system. It is responsible for analyzing the uploaded content to determine whether it is harmful or not. For images, OpenCV is used to read and process the file. The image is converted into grayscale to simplify analysis, and the average brightness is calculated. This brightness value is used as an indicator of potentially harmful content. The module is designed to be fast and efficient, enabling real-time processing without requiring heavy computational resources

G. Decision-Making Module

The decision-making module receives input from the content analysis module and determines whether the content should be allowed or blocked. It uses a threshold-based approach, where the calculated brightness value is compared with a predefined limit. If the value exceeds the threshold, the content is considered harmful and is rejected. Otherwise, it is accepted. This module ensures quick decision-making and forms the core logic of the system.

H. Blocking Mechanism

If harmful content is detected, the blocking mechanism is activated. This module immediately deletes the uploaded file from the system and prevents it from being stored or displayed. It also sends a message to the user indicating that the content is harmful. This ensures that inappropriate content does not reach other users, maintaining the safety of the platform.

I. Storage Module

If the content is considered safe, it is stored in a designated folder (static/uploads). This module manages file storage and ensures that files are organized properly. It also supports easy retrieval of files for display. The storage system is simple and efficient, making it suitable for small-scale application.

IV. METHODOLOGY

The methodology of the proposed **Automated Harmful Content Control and Blocking System** focuses on developing a structured approach to detect and prevent harmful content before it is uploaded. The system follows a step-by-step process that includes data input, validation, processing, analysis, decision-making, and output generation. The primary objective is to ensure real-time detection using lightweight image processing techniques.

A. System Design and Approach

The system is designed using a modular approach, where each component performs a specific function. This modular

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.13511

structure improves maintainability, scalability, and ease of implementation. The methodology follows a **client-server architecture**, where the client (user interface) interacts with the server (Flask backend) for processing requests.

B. Data collection and input handling

The first step in the methodology involves collecting input from the user. The system provides a web-based interface where users can upload images or videos. The uploaded file is captured using an HTML form and transmitted to the backend using the POST method with multipart encoding. This ensures secure and efficient data transfer.

C. File Processing

Once the file is received, preprocessing is performed to prepare it for analysis. This includes:
Checking if the file is empty
Extracting the filename and file extension
Converting the filename into a secure format using safe naming techniques:
Generating the file path for storage
Preprocessing ensures that the file is ready for validation and analysis.

D. File Validation

In this stage, the system verifies whether the uploaded file meets the required criteria. The file extension is checked against a predefined list of allowed formats such as .jpg,
.png, .jpeg, .mp4, .avi, and .mov. If the file does not meet the criteria, it is rejected immediately, and the user is notified. This step ensures system security and prevents unsupported files from being processed.

E. File Storage

After validation, the file is temporarily saved in the system's storage directory (static/uploads). This allows the system to access the file for further processing. Temporary storage ensures that the file can be analyzed before making a final decision.

F. Content Analysis

The core part of the methodology is the content analysis phase. If the uploaded file is an image, the system uses OpenCV to analyze its visual properties. The image is first read using OpenCV functions and then converted into grayscale format.Grayscale conversion simplifies the image by reducing it to a single intensity channel, making it easier to analyze. The system then calculates the average brightness of the image by computing the mean pixel intensity value.The brightness value is used as a simple indicator of abnormal or potentially harmful content. This approach is computationally efficient and suitable for real-time processing.

G. Decision Making Process

After calculating the brightness value, the system applies a threshold-based decision rule. A predefined threshold value is used to classify the image as either safe or harmful.If the brightness value exceeds the threshold, the image is considered harmful If the brightness value is within the acceptable range, the image is considered safe This decision-making process is fast and ensures immediate response to user uploads. It forms the core logic of the system and enables real-time moderation.

H. Blocking Mechanism

If the system identifies the content as harmful, the blocking mechanism is triggered. The uploaded file is immediately deleted from the storage folder, preventing it from being stored or displayed. The system also generates an alert message to inform the user that the content has been blocked due to harmful characteristics.This proactive approach ensures that harmful content is stopped before it becomes accessible to others, improving overall platform safety.

I. Safe Content Handling

If the content is classified as safe, the system allows the file to remain in the storage folder. The file is then made available for display on the user interface. This ensures that only safe and appropriate content is stored and shared.The system also maintains a list of uploaded files, which can be accessed and viewed by the user. This provides transparency and usability.

J. Video Handling

For video files, the system performs only basic validation and storage. Since video analysis requires complex processing, the current implementation does not perform detailed content analysis on videos. However, the system ensures that video files are uploaded correctly and stored without errors.This provides flexibility for future

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.13511

enhancements, where advanced video analysis techniques can be integrated.

K. Display and User Interaction

The system retrieves stored files and displays them on the web interface. Users can view the list of uploaded files along with relevant options. The interface is designed to be simple and easy to use, ensuring smooth interaction between the user and the system.

L. Deletion Process

The system provides a delete option that allows users to remove uploaded files. When the delete request is triggered, the system identifies the file and removes it from the storage location. The interface is then updated to reflect the changes.This feature enhances user control and ensures proper content management.

V. SYSTEM IMPLEMENTATION

The System Implementation phase focuses on the practical development of the proposed Automated Harmful Content Control and Blocking System. It involves converting the system design into a working application using appropriate technologies and tools. The implementation ensures that all modules such as file upload, validation, content detection, and deletion are integrated and function correctly in real time.

The system is developed using Python as the programming language due to its simplicity and extensive library support. The backend is implemented using Flask, a lightweight web framework, while OpenCV is used for image processing and analysis. The frontend is designed using HTML to provide a simple and user-friendly interface.

A. Development Environment

The system is developed and tested in a standard development environment with the following tools:

Programming Language: Python
Framework: Flask
Libraries: OpenCV, NumPy, Werkzeug
Frontend: HTML
Editor: Visual Studio Code
Operating System: Windows

Python is chosen because it supports rapid development and has strong libraries for image processing and web development. Flask is used due to its lightweight nature and ease of integration

B. Backend Implementation

The backend is the core part of the system and is responsible for processing user requests and performing content moderation.

Flask Application Setup File Upload Handling File Validation
Secure File Storage

C. Image Processing Implementation

The system uses OpenCV for analyzing image content. Image Reading
Grayscale Conversion Brightness Calculation Threshold-Based Detection

D. Blocking Mechanism Implementation

If harmful content is detected, the system immediately deletes the file:
os.remove(filepath)
An alert message is displayed to inform the user. This ensures that harmful content is not stored or displayed.

E. Video Handling Implementation

For video files, the system currently performs validation and storage only. Since video processing is computationally intensive, detailed analysis is not implemented in the current version.

However, the system supports uploading video files and ensures they are saved correctly without errors. This allows future integration of advanced video analysis techniques.

F. Frontend Implementation

The frontend is developed using HTML and provides a simple interface for users.
File Upload Form Display of Uploaded Files Delete Option

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.13511

G. Delete Function Implementation

The delete functionality is implemented in Flask:

```
@app.route('/delete/<filename>')
def delete_file(filename):
path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
if os.path.exists(path):
os.remove(path)
```

This ensures proper file removal from the system.

H. Error Handling

The system includes error handling mechanisms to ensure smooth operation:
Handles missing files Handles invalid file formats
Handles image processing errors Displays user-friendly messages
This improves reliability and user experience

I. Integration of Modules

All modules are integrated to work together seamlessly. The frontend sends requests to the backend, which processes the data and returns results. The system ensures smooth communication between all components.

J. Testing and Execution

The application is tested by running the Flask server:

```
python app.py
```

The system is accessed through:

http://127.0.0.1:5000
Different test cases are used, including:

Uploading valid images Uploading harmful images Uploading videos Deleting files

K. Performance Evaluation

The system performs efficiently due to its lightweight design. Image processing is fast, and decisions are made in real time. The system works well even on low-resource machines.

VI. EXPERIMENTAL RESULTS AND PERFORMANCE EVALUATION

The Experimental Results and Performance Evaluation section presents the outcomes obtained after implementing the Automated Harmful Content Control and Blocking System. This section evaluates how effectively the system performs in detecting harmful content, handling file uploads, and providing real-time responses. The system is tested under various conditions to analyze its accuracy, efficiency, and reliability.where Pi is the actual price and Pˆi is the predicted price.For demand forecasting, the accuracy of the forecast was measured by comparing the predicted demand values with the actual market arrival data using RMSE.

A. Experimental Setup

The system was tested in a controlled environment using a standard personal computer. The implementation was carried out using Python and Flask as the backend framework, with OpenCV for image processing.

System Configuration:

Operating System: Windows Programming Language: Python Framework: Flask
Libraries: OpenCV, NumPy Interface: Web browser (localhost)
The application was executed on a local server using the command:

```
python app.py
```

The system was accessed through the browser using:

http://127.0.0.1:5000
Various test cases were considered to evaluate the performance of the system.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.13511

B. Test Cases and Observations

The system was tested with different types of inputs to verify its behavior:

Uploading a Valid Image Uploading a Harmful Image Uploading Unsupported File Uploading a Video File

C. Performance Metrics

The system performance is evaluated based on the following parameters:

Accuracy:

The system correctly identifies harmful and non-harmful images based on brightness threshold. Although the method is simple, it provides acceptable accuracy for basic detection.

Response Time:

The system processes uploads in real time. The average response time is very low since image processing is lightweight.

Efficiency:

The system efficiently handles multiple uploads without significant delay. The use of simple algorithms reduces computational overhead.

Reliability:

The system consistently performs the required operations such as upload, detection, and deletion without errors.

Usability:

The user interface is simple and easy to use, allowing users to interact with the system without technical knowledge.

D. Result Analysis

The experimental results show that the system successfully performs real-time content moderation. Harmful images are effectively blocked before being stored, which ensures platform safety. The system also handles video uploads correctly and provides a functional delete feature.

However, the detection mechanism is based on brightness, which may not accurately identify all types of harmful content. Despite this limitation, the system achieves its primary goal of demonstrating pre-upload content filtering.

E. Comparative Analysis

Compared to traditional systems that perform post-upload moderation, the proposed system offers significant advantages:

Prevents harmful content before it is published Provides faster response time

Requires less computational power Offers user-level control through deletion

F. Limitations Observed During Testing

Detection is limited to brightness-based analysis Cannot identify complex harmful patterns Video content is not analyzed deeply

May produce false positives in some cases

G. Overall Performance Evaluation

Overall, the system performs efficiently in real-time scenarios. It successfully demonstrates the concept of proactive content moderation by blocking harmful content before upload. The system is lightweight, easy to implement, and suitable for small-scale applications.

H. Summary

The experimental results confirm that the proposed system is capable of:

Detecting harmful images Blocking inappropriate content Supporting video uploads Providing file management features

This validates the effectiveness of the system in achieving its object.

VII. DISCUSSION

The proposed Automated Harmful Content Control and Blocking System effectively demonstrates a real-time approach to content moderation. The system is capable of analyzing uploaded images before they are stored and successfully blocks content that is identified as harmful. This proactive mechanism helps in preventing the spread of inappropriate material and ensures a safer platform for users.

The system performs efficiently due to its lightweight design and simple image processing technique. The brightness-based detection method allows quick decision-making with minimal response time. Additionally, the user interface is

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IARJSET.2026.13511

easy to use, enabling users to upload files, view content, and delete files without difficulty. The system also supports video uploads, showing its flexibility for handling different types of media.

However, the system has certain limitations. The detection method is based only on brightness, which may not accurately identify all types of harmful content and can sometimes produce incorrect results. Also, video files are not deeply analyzed in the current implementation. Despite these limitations, the system provides a strong foundation for real-time content moderation and can be enhanced in the future using advanced techniques like machine learning.

VIII. CONCLUSION AND FUTURE WORK

The Automated Harmful Content Control and Blocking System presents an effective approach for real-time moderation of user-generated content. The system successfully analyzes uploaded images before storage and prevents harmful content from being published. By using a Flask-based backend and OpenCV for image processing, the system achieves fast performance with low computational requirements. Features such as file validation, user feedback, and deletion functionality improve usability and provide better control over uploaded content. The implementation demonstrates that a lightweight system can efficiently handle basic content moderation tasks.

However, the current system uses a simple brightness-based detection method, which may not accurately identify all types of harmful content. It also does not perform detailed analysis of video files. These limitations highlight the need for further improvements to enhance accuracy and functionality.

In future work, the system can be extended by integrating advanced machine learning and deep learning techniques such as Convolutional Neural Networks (CNNs) for more accurate detection. Video analysis can be improved by implementing frame-based processing methods. Additional features such as user authentication, behavior tracking, and cloud storage can enhance scalability and security. With these enhancements, the system can evolve into a more robust and practical solution for real-world content moderation.platform.

REFERENCES

[1]. CLARITY, “A Lightweight Cross-Modal Transformer for Harmful Content Detection,” *IEEE Transactions on Artificial Intelligence*, vol. 7, no. 2, pp. 120–130, 2026.

[2]. Li, X., Wang, Y., and Chen, Z., “Dual-Branch Neural Network for Harmful Meme Detection,” *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 450–460, 2025.

[3]. Kumar, A., Singh, R., and Patel, S., “MAHGA: Multi-Aspect Heterogeneous Graph Analysis for Harmful Speech Detection,” *IEEE Access*, vol. 13, pp. 22045–22060, 2025.

[4]. Smith, J., Brown, L., and Davis, M., “Supervised Learning Approach for Sensitive Content Classification in Social Media,” *Journal of Artificial Intelligence Research*, vol. 68,pp. 789–805, 2024.

[5]. Zhang, Y., Liu, H., and Zhao, Q., “MTikGuard: Transformer-Based Multimodal Moderation System for Short Video Platforms,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1023–1032, 2024.

[6]. I. Goodfellow et al., “Explaining and Harnessing Adversarial Examples,” *International Conference on Learning Representations (ICLR)*, 2015.

[7]. K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[8]. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

[9]. M. Tan and Q. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 6105–6114, 2019.

[10]. OpenCV, “Open Source Computer Vision Library,”Available: https://opencv.org/

[11]. Flask Documentation, “Flask Web Framework,” Available: https://flask.palletsprojects.com/

 This work is licensed under a Creative Commons Attribution 4.0 International License