IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2025

DOI: 10.17148/IJIREEICE.2025.13513

AI DRIVEN SOLUTION FOR ETHICAL TEXT AND IMAGE MODERATION

Dr. Akshath M J[1], Deekshitha C[2], Deekshitha M[3], Deepika T R[4], Deepthi K[5]

Associate Professor, Department of CS&E, Maharaja Institution of Technology Mysore, Mandya, India[1]

Student, Department of CS&E, Maharaja Institution of Technology Mysore, Mandya, India[2]

Student, Department of CS&E, Maharaja Institution of Technology Mysore, Mandya, India[3]

Student, Department of CS&E, Maharaja Institution of Technology Mysore, Mandya, India[4]

Student, Department of CS&E, Maharaja Institution of Technology Mysore, Mandya, India[5]

Abstract: This project proposes an AI-powered solution for moderating unethical and harmful text and image content on digital platforms. With the surge in user-generated content, challenges like hate speech, explicit visuals, and cyberbullying require real-time, accurate, and ethical moderation. The system utilizes Google Cloud's Natural Language API and image recognition tools to automatically detect, classify, and flag inappropriate content. It combines advanced NLP and computer vision techniques to support multilingual input and context-aware analysis. The integrated administrator dashboard enhances review efficiency, while Explainable AI ensures transparency. This framework aims to create safer online spaces through scalable and ethical content governance.

Keywords: AI-powered solution, Content moderation, Natural Language API, Image recognition tools, NLP (Natural Language Processing), Computer vision, Multilingual input, Explainable AI, Administrator dashboard, Ethical content governance.

I. INTRODUCTION

1.1 Overview The exponential growth of user-generated content across digital platforms has raised critical concerns about online safety and ethical consumption. The pervasive presence of harmful material, including hate speech, explicit visuals, and misinformation, necessitates robust content moderation. This project introduces an AI-driven solution designed to automatically filter and flag unethical text and image content in real time, ensuring adherence to community guidelines and ethical standards.

1.2 Problem Statement Current online environments lack efficient, real-time, and ethical mechanisms for moderating harmful text and image content, compromising user safety and community integrity.

1.3 Solution Our proposed system leverages advanced AI techniques, including Google Cloud's Natural Language API and image moderation tools, to detect and classify offensive or inappropriate content. It alerts administrators for review, thereby promoting higher ethical content standards online.

1.4 Existing System Existing moderation methods, often relying on manual review or keyword filtering, are inefficient and prone to contextual errors. While some platforms use hybrid AI-human approaches, they still exhibit limitations in real-time accuracy and consistency for complex content. Our AI-driven solution significantly enhances these systems by providing more precise, automated, and context-aware content analysis.

II. LITERATURE SURVEY

A. Literature Review

1. **Overview of AI in Content Moderation:** AI has revolutionized online content moderation by enabling rapid, large-scale analysis of user-generated content for safer digital spaces. Key techniques include Natural Language Processing (NLP) for textual content (e.g., detecting hate speech using Transformer models), Computer Vision for visual content (e.g., identifying explicit material via CNNs), and Multimodal Analysis, which combines both to understand complex violations in integrated text-image content like memes.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2025

DOI: 10.17148/IJIREEICE.2025.13513

2. **Challenges in Existing Systems:** Despite advancements, AI moderation faces challenges. Contextual misunderstandings lead to misinterpretations of satire or cultural expressions. Biased training data can result in unfair moderation outcomes, disproportionately affecting marginalized groups. Lack of transparency in "black box" AI systems erodes user trust. The immense volume of content demands significant scalability, and adversarial attacks constantly challenge detection capabilities.

3. **Emerging Trends and Solutions:** To address these challenges, researchers are exploring Explainable AI (XAI) for transparency, hybrid models combining diverse AI techniques for robustness, and ethical AI frameworks to mitigate bias. Human-in-the-loop systems integrate human oversight for complex cases, while federated learning and localized models enhance scalability and linguistic diversity.

B. Survey Findings

A review of current practices confirms AI's crucial role in content moderation for its automation and efficiency. Platforms like Google and Meta utilize advanced NLP and computer vision. However, contextual issues and biases persist, driving a need for continuous innovation. Best practices include hybrid moderation (AI + human), prioritizing privacy and ethics, continuous model updates, and user-centric design with clear explanations and appeal processes. Future trends point towards scalable, ethical, and transparent AI systems capable of high-accuracy multimodal content processing.

III. SOFTWARE REQUIREMENT SPECIFICATIONS

A. ***Functional Requirements*** : The system's functional requirements define its core capabilities for processing, evaluating, and managing user-generated content, ensuring robust administrative oversight.

1. **Text Moderation Functionality:** Accepts text, uses Google Cloud Natural Language API for real-time sentiment analysis, entity recognition, and toxicity detection. Classifies text as Safe, Offensive, Profane, or Harmful with a confidence score.

2. **Image Moderation Functionality:** Processes images, uses pre-trained models (e.g., Google Model Garden) to detect explicit or harmful content, flagging problematic images for administrator review.

3. **Real-time Monitoring:** Continuously scans incoming content, triggering immediate actions like administrator alerts or temporary hiding based on severity. Supports dynamic content streams with minimal latency.

4. **Administrator Dashboard:** Provides a centralized web interface for reviewing flagged content, accessing reports, and making decisions. Features include a prioritized queue, manual override tools, decision logs, and visual analytics.

5. **Multilingual Content Handling:** Leverages Google's multilingual NLP for moderation across various languages (English, Spanish, Mandarin, Arabic, Hindi), accounting for cultural nuances.

6. **Reporting System:** Users can report content via an in-platform tool. Reported content is prioritized for admin review, with metadata capturing the rationale, ensuring timely resolution.

B. ***Non-Functional Requirements*** : These requirements establish quality attributes, performance benchmarks, and operational constraints to ensure system reliability, scalability, and user satisfaction.

- **Performance:** Processes content within 1-2 seconds, handling at least 10,000 checks daily, with low latency via optimized API calls and efficient model inference.
- **Scalability:** Designed for horizontal scaling across cloud infrastructure using containerized microservices and auto-scaling mechanisms.
- **Accuracy:** Aims for over 90% accuracy in detecting harmful content, minimizing false positives and negatives through continuous model retraining and human feedback.
- **Security:** Secures data transmission via HTTPS with TLS 1.3, protects user privacy through anonymization and compliance, and uses role-based access controls.
- **Reliability:** Guarantees 99.9% uptime, logs all decisions in a tamper-proof audit trail, and uses redundant systems for minimal disruption.
- **Usability:** Features an intuitive administrator dashboard with responsive design, customizable filters, and simplified user reporting tools.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2025

DOI: 10.17148/IJIREEICE.2025.13513

- **Compliance:** Adheres to ethical and legal guidelines (e.g., COPPA, EU's DSA), incorporating fairness checks and regular audits.

C. ***System Requirements:***

TABLE I : HARDWARE REQUIREMENTS

Component	Specification
Processor	Intel i5 or higher
RAM	8 GB minimum
Storage	250 GB HDD or SSD
Network	Stable internet connection for API integration

TABLE II: SOFTWARE REQUIREMENTS

Software	Purpose
Operating System	Windows 10 / Linux / MacOS
Frontend	HTML, CSS, JavaScript
Backend	Python (Flask/Django)
APIs	Google Cloud Natural Language API, Google Model Garden, Eden AI API
Database	MySQL / Firebase / MongoDB
Browser	Chrome / Firefox (for testing UI)
Tools	Postman (API testing), Visual Studio Code (IDE), GitHub (version control)

IV. SYSTEM ANALYSIS AND DESIGN

A. System Analysis

1. **Feasibility Study:** This study assesses the project's practicality, confirming its technical, economic, and operational viability within constraints.
   - *Technical Feasibility:* Utilizes established cloud AI tools (Google Cloud Natural Language API, Google Model Garden) via REST APIs. Standard development languages (Python, JavaScript) and frameworks (Flask) ensure robust handling. Deployment is supported by containerization (Docker) and orchestration (Kubernetes), ensuring scalability and reliability. Minimal local hardware is required.
   - *Economic Feasibility:* Leverages cost-effective cloud AI services with free tiers and pay-per-use models. Open-source tools and frameworks reduce licensing costs, making the project affordable for academic or prototype purposes.
   - *Operational Feasibility:* Designed for ease of use with an intuitive admin interface. Training is minimal. Integration with existing platforms is achievable. Automated moderation reduces workload, while manual overrides offer flexibility.
2. **SWOT Analysis:** The SWOT analysis evaluates the system's internal strengths and weaknesses alongside external opportunities and threats.

.

TABLE III: SWOT ANALYSIS OF THE AI-BASED CONTENT MODERATION SYSTEM

Strengths	Weaknesses
Employs advanced AI models (Transformers, CNNs) for accurate text and image moderation.	Struggles with nuanced content like sarcasm, cultural idioms, or regional slang, risking misclassification.
Supports real-time content scanning, enabling prompt responses to policy violations.	Depends on reliable internet for API calls, potentially causing delays in low-connectivity settings.
Integrates seamlessly with cloud platforms, ensuring ease of deployment and maintenance.	Requires technical expertise for initial API setup and configuration.
Opportunities	**Threats**

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2025

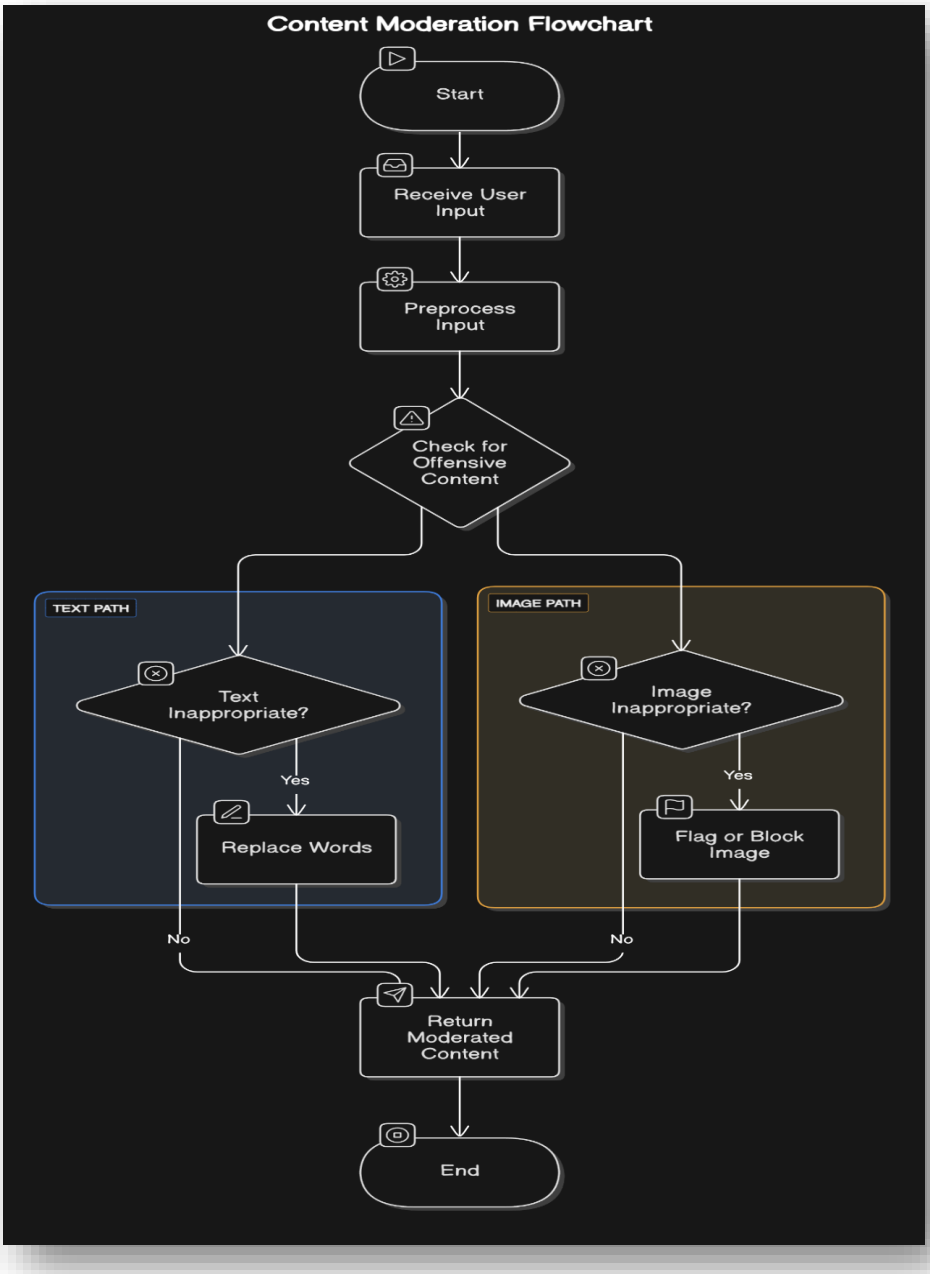
DOI: 10.17148/IJIREEICE.2025.13513

Applicable to diverse sectors (social media, education, gaming), broadening utility.	Increasing API costs with higher content volumes may challenge budget scalability.
Can adopt emerging AI advancements (Explainable AI) to improve transparency and accuracy.	Rapidly changing content trends (new slang, manipulated images) may outdate models.
Meets growing demand for compliant moderation in regulated industries, opening niche markets.	Stricter regulations may require system updates, increasing development costs.

The SWOT analysis highlights the system's robust technical foundation and market potential while identifying challenges in handling nuanced content and managing costs.

B. System Design :

1. **Flowchart:**



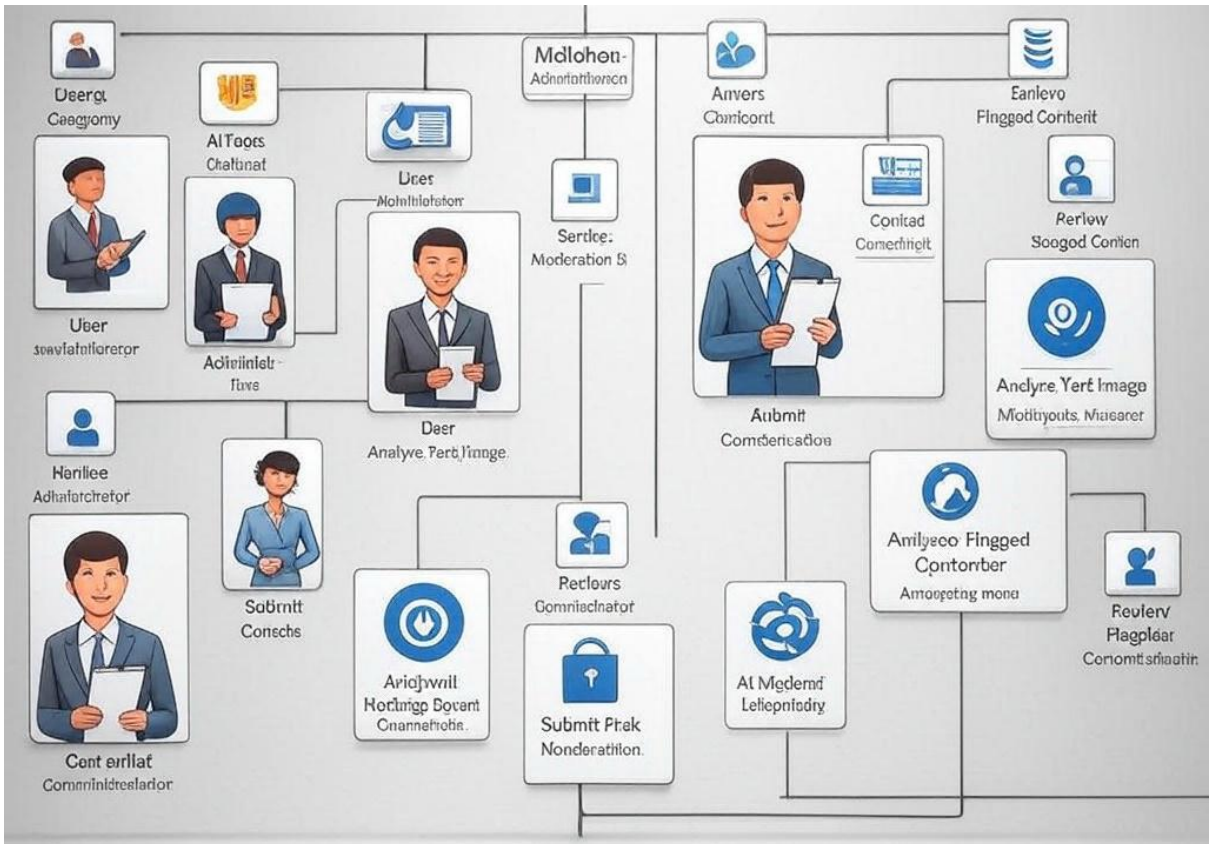
IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2025

DOI: 10.17148/IJIREEICE.2025.13513

2. **Use Case Diagram**:



V. IMPLEMENTATION

A. Implementation Strategy

The project's implementation follows a modular approach, integrating frontend, backend, and cloud-based AI services for both text and image content moderation. This strategy ensures simple integration, high accuracy, and a responsive user experience.

B. Modules of Implementation

1. **User Input Interface:** HTML, CSS, JavaScript for collecting user-generated text and image uploads. Features a simple form and real-time feedback.

2. **Backend Server:** Python with Flask, responsible for receiving user input, forwarding it to relevant APIs (Google Cloud NLP, Eden AI/Image Moderation), and returning moderation decisions. Handles REST API requests.

3. **Text Moderation Engine:** Utilizes Google Cloud Natural Language API to analyze text for offensive, violent, profane, and harmful language, returning sentiment, entity classification, and content category.

4. **Image Moderation Engine:** Employs Google Model Garden / Eden AI Image Moderation to detect nudity, explicit imagery, weapons, and hate symbols, labeling content with severity levels.

5. **Admin Dashboard:** Flask (Backend) and HTML/CSS/JS (Frontend) provide an interface to view flagged content details, manually approve/reject content, and access moderation history and user reports.

6. **Database:** SQLite / MySQL for storing user information, content data with moderation results, user-reported content, and admin decisions.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2025

DOI: 10.17148/IJIREEICE.2025.13513

C. API Integration

1. **Text Moderation API Integration:** Integrates with https://language.googleapis.com/v1/documents:analyzeContent. The payload includes user text, language, and encoding, returning content classification, sentiment, and categories.

2. **Image Moderation API Integration:** Integrates with Eden AI / Google Model Garden endpoints. The payload is a Base64-encoded image or image URL, returning the probability of unsafe content and labels (e.g., NSFW, Weapon, Gore).

D. Sample Code Snippet (JSON file)

```
// Load JSON (Replace with actual JSON file if needed)
const jsonData = {
  "sensitive_regions": [
  {
    "type": "blood",
    "coordinates": { "x": 200, "y": 200, "width": 200, "height": 300 },
    "description": "Blood splatters on clothing and ground"
  },
{
    "type": "violence",
    "coordinates": { "x": 250, "y": 250, "width": 150, "height": 250 },
    "description": "Character covered in blood, holding a weapon"
}
]
};
Each entry must have a clear type and a concise description.`,
  {
    fileData: {
       fileUri: uploadResult.file.uri,
       mimeType: uploadResult.file.mimeType,
     },
    },
]);
// Output the result from the AI response
// console.log(filter(result.response.text()));
return filter(result.response.text());
}
// run();
export default run;
```

index.js

```
import express from 'express';
import multer from 'multer';
import * as Jimp from 'jimp'; // Import everything as an object
import { v4 as uuidv4 } from 'uuid';
import fs from 'fs';import path from 'path';
import bodyParser from 'body-parser';
import run from './run.js';
import run2 from './run2.js';
import blur from './blur.js';
import { dirname } from 'path';
import { fileURLToPath } from 'url';
import { log } from 'console';

const __filename = fileURLToPath(import.meta.url);
const __dirname = dirname(__filename);

const app = express();
const port = 3000;

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));

// Set up Multer for file uploads
const storage = multer.diskStorage({
   destination: './public/uploads/', // Folder where images will be stored
filename: (req, file, cb) => {
   const uniqueName = uuidv4() + path.extname(file.originalname); // Generate UUID
   cb(null, uniqueName);
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2025

DOI: 10.17148/IJIREEICE.2025.13513

```
}
});

const upload = multer({ storage: storage });

const fileFilter = (req, file, cb) => {
if (file.mimetype.startsWith('image/')) {
   cb(null, true);
} else {
   cb(new Error('Not an image'), false);
   }
};
// Serve static files from 'public'
app.use(express.static('public'));

// Parse form data
app.use(express.urlencoded({ extended: true }));
app.use(express.json());

// Route for text moderation
app.post('/moderate-text', async(req, res) => {
const text = req.body.text;
console.log(req.body);
if (!text) {
   return res.status(400).json({ error: 'Text is required' });
}

const moderatedText = await run(text);
console.log(moderatedText);
res.json({ text: JSON.parse(moderatedText) });
});

// Route for image moderation
app.post('/moderate-image', upload.single('image'), async(req, res) => {
if (!req.file) {
   return res.status(400).send('No file uploaded.');
}

const path="C:/Users/HP/OneDrive/Desktop/cont/public/uploads/"+req.file.filename;
console.log(path);
const result=(await run2(path));
console.log("============================\n"+result);
await blur(path,'public/images/'+req.file.filename,JSON.parse(result).sensitive_regions);
// console.log(newpath);
res.json({
generated_image: req.file.filename,
"results":JSON.parse(result).sensitive_regions
});
});

async function deleteFilesInFolder(folderPath) {
try {
   const files = await fs.promises.readdir(folderPath);
   await Promise.all(files.map(file => fs.promises.unlink(path.join(folderPath, file))));
   // console.log('All files deleted.');
} catch (err) {
   console.error('Error:', err);
   }
}

// Start the server
app.listen(port, async() => {

    await deleteFilesInFolder(path.join(__dirname, 'public/images'));
    await 3
    3
    deleteFilesInFolder(path.join(__dirname, 'public/uploads'));

    console.log(`Server running at http://localhost:${port}`);
});
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2025

DOI: 10.17148/IJIREEICE.2025.13513

blur.js

```
import sharp from 'sharp';
import fs from 'fs';

// Load JSON (Replace with actual JSON file if needed)
const jsonData = {
   "sensitive_regions": [
  {
     "type": "blood",
     "coordinates": { "x": 200, "y": 200, "width": 200, "height": 300 },
     "description": "Blood splatters on clothing and ground"
},
{
     "type": "violence",
     "coordinates": { "x": 250, "y": 250, "width": 150, "height": 250 },
     "description": "Character covered in blood, holding a weapon"
   }
  ]
};

async function blurSensitiveRegions(inputPath, outputPath, sensitiveRegions, blurAmount =
100, scale = 1.5) {
   // Get image dimensions
   const { width: imgWidth, height: imgHeight } = await sharp(inputPath).metadata();

   Let image = sharp(inputPath);

  // Process each region with scaling
   const overlays = await Promise.all(sensitiveRegions.map(async (region) => {
     let { x, y, width, height } = region.coordinates;

   // Scale coordinates
   let scaledX = Math.round(x * scale);
   let scaledY = Math.round(y * scale);
   let scaledWidth = Math.round(width * scale);
   let scaledHeight = Math.round(height * scale);

   // Ensure values stay within image bounds
   scaledX = Math.max(0, Math.min(scaledX, imgWidth - 1));
   scaledY = Math.max(0, Math.min(scaledY, imgHeight - 1));
   scaledWidth = Math.max(1, Math.min(scaledWidth, imgWidth - scaledX));
   scaledHeight = Math.max(1, Math.min(scaledHeight, imgHeight - scaledY));

   try {
      // Extract and blur the specific region
      const blurredRegion = await image.clone()
         .extract({ left: scaledX, top: scaledY, width: scaledWidth, height: scaledHeight })
         .blur(blurAmount)
         .toBuffer();

   return { input: blurredRegion, left: scaledX, top: scaledY };
   } catch (error) {
      console.error(`Error processing region: ${JSON.stringify(region.coordinates)}`, error);
      return null;
   }
}));

// Remove null values in case of errors
const validOverlays = overlays.filter(overlay => overlay !== null);

if (validOverlays.length > 0) {
   await image.composite(validOverlays).toFile(outputPath);
console.log("Blurred sensitive regions saved to:", outputPath);
} else {
     console.error("No valid regions to process. Check image dimensions and JSON input.");
   }
}

export default blurSensitiveRegions;
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2025

DOI: 10.17148/IJIREEICE.2025.13513

E. Hosting and Testing

- Hosting: Flask app hosted on localhost (or Google Cloud Run for deployment).
- Testing Tools:
  - o Postman (API testing)
  - o Browser console for frontend validation
  - o Logs for moderation results

VI. TESTING AND RESULTS

A. Testing Strategy

A combination of manual testing and automated test cases was used to validate the functionality, accuracy, and performance of the system. Testing focused on ensuring that both text and image inputs were properly analyzed and flagged according to ethical standards.

B. Types of Testing Performed

Table 1: Summary of Testing Types Applied to the Moderation System

Test Type	Description
Unit Testing	Tested individual modules like text input, image upload, and API calls.
Integration Testing	Ensured backend correctly integrates with Google NLP and Eden AI services.
System Testing	Verified that the entire moderation workflow performs as expected.
Usability Testing	Checked user interface clarity and admin dashboard ease of use.
Performance Testing	Monitored speed of moderation response under various loads.
Boundary Testing	Tested with extremely short/long texts and high-resolution images.
Error Handling Tests	Simulated invalid inputs and API failures to ensure graceful error recovery.

Sample Test Case

Table 2: Functional Test Cases for Text and Image Moderation System

Test Case ID	Test Case Description	Input	Expected Output	Result
TC01	Offensive Text Detection	"You are an idiot"	Flagged as offensive	Passed
TC02	Safe Text Detection	"Have a nice day!"	Approved	Passed
TC03	Image with Nudity	NSFW image	Flagged as explicit	Passed
TC04	Image with No Offensive Content	Landscape image	Approved	Passed
TC05	Missing Image File	No file submitted	Error: Image not found	Passed
TC06	Non-English Content	"Mar jaa" (Hindi abuse)	Flagged	Passed
TC07	Real-Time Feedback	Fast API response	< 2 seconds	Passed

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2025

DOI: 10.17148/IJIREEICE.2025.13513

C. **Test Environment Setup**

- **Device**: Laptop with 8 GB RAM and Intel i5 processor
- **Browser**: Google Chrome
- **Backend**: Flask Server on localhost
- **APIs Used**:
  - Google Natural Language API
  - Eden AI Image Moderation API
- **Tools**:
  - Postman (API testing)
  - Google Cloud Console (API monitoring)

D. **Results and Observations**

- **Accuracy**:
  - Text moderation performed well on profanity, hate speech, and threats.
  - Image moderation accurately flagged explicit and harmful visuals.
- **Response Time**:
  - Average response time for moderation: **1.7 seconds**.
  - Real-time moderation worked without lag in testing scenarios.
- **User Experience**:
  - Simple, intuitive frontend for users.
  - Admin dashboard allowed easy manual moderation.
- **Limitations Noticed**:
  - Some context-sensitive sarcasm wasn't always detected.
  - Multilingual moderation relied on API language auto-detection.

D. **Screenshots**

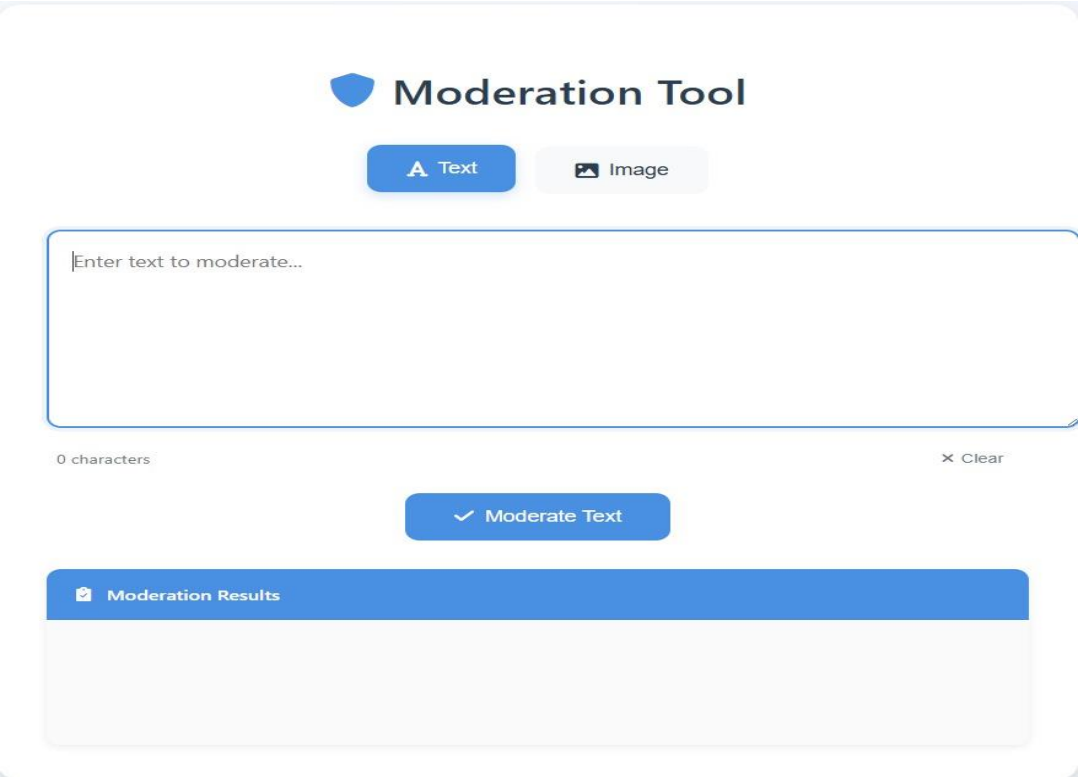


fig. 1 User Interface for Text Moderation

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2025

DOI: 10.17148/IJIREEICE.2025.13513

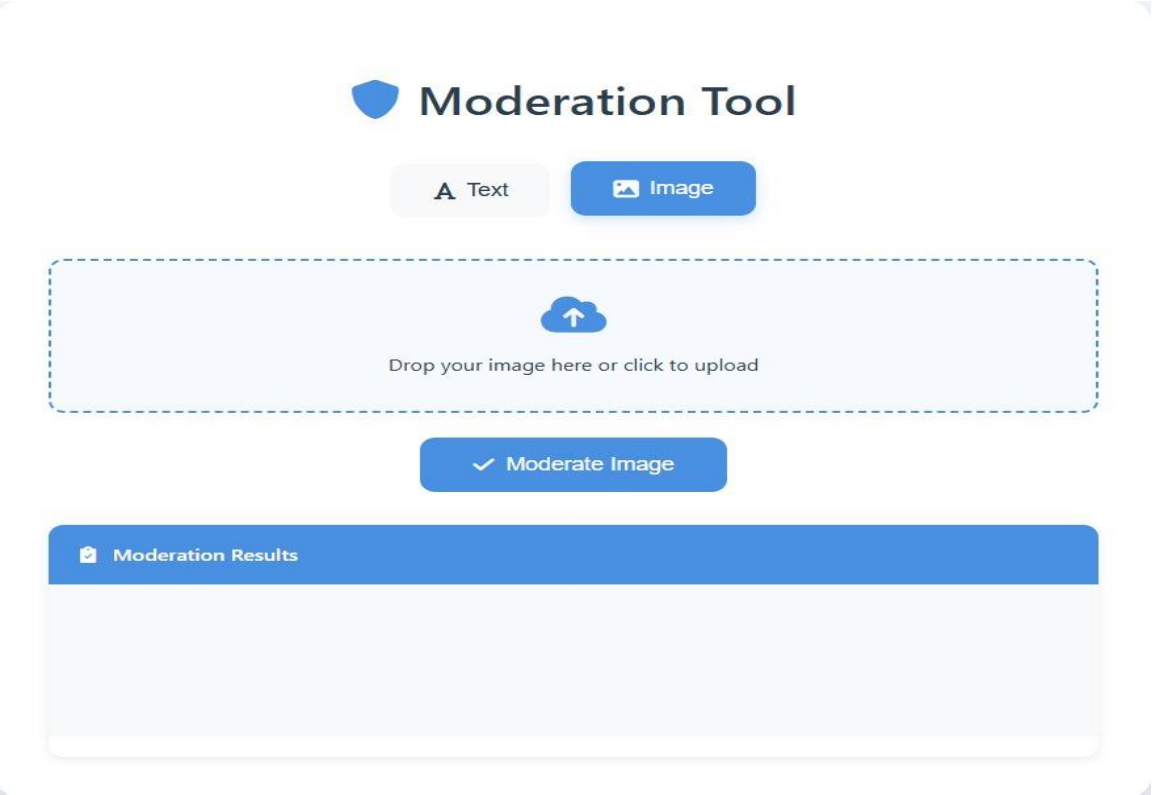


fig. 2 User Interface for Image Moderation

VII. CONCLUSION

The project titled **"AI-Driven Solution for Ethical Text and Image Moderation"** successfully demonstrates the potential of Artificial Intelligence in ensuring safe, inclusive, and responsible digital environments. By integrating advanced technologies such as **Natural Language Processing**, **Computer Vision**, and **Explainable AI**, the system automates the detection and classification of unethical content in both text and images.

The framework effectively handles multilingual input, processes real-time content, and provides administrators with a user-friendly dashboard for oversight. Its modular and scalable design allows for easy integration across various digital platforms, making it adaptable to diverse applications such as social media, online learning, and forums.

This solution significantly reduces the reliance on manual moderation while improving accuracy, speed, and transparency. As the digital landscape continues to evolve, such intelligent systems are essential for maintaining ethical standards and enhancing user trust.

VIII. FUTURE ENHANCEMENTS

While the current system performs well, several improvements can expand its capabilities:

1. **Multilingual Support**
   Add more languages (e.g., Kannada, Arabic, Swahili) and use language-specific NLP models for better cultural understanding.
2. **Custom AI Models**
   Train models on domain-specific data (gaming, healthcare) to detect niche violations like medical misinformation or sarcasm.

DOI: 10.17148/IJIREEICE.2025.13513

3. **Video & Audio Moderation**
   Support moderation of videos and audio using frame analysis and speech-to-text for detecting verbal abuse or inappropriate sounds.
4. **User Reputation Tracking**
   Implement behavior scoring to identify repeat offenders and reward ethical contributors. Use analytics to spot abuse trends.
5. **Advanced Dashboard Features**
   Add visual tools like heatmaps and trend reports for better admin insights and decision-making.
6. **Offline Moderation**
   Build lightweight models for use in low-connectivity environments, optimized for mobile and edge devices.
7. **Legal & Ethical Compliance**
   Ensure compliance with laws like GDPR and DSA using privacy-first design and regular audit support.

These upgrades can transform the system into a robust, global-ready moderation platform.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to the **Department of Computer Science and Engineering, MITM**, for providing the necessary resources and support to carry out this project successfully.

We extend our heartfelt thanks to our project guide, **Dr. Akshath M J** (Associative Professor), for their continuous guidance, valuable insights, and encouragement throughout the development of this work.

We also thank our **Head of Department, faculty members, and technical staff** for their cooperation and feedback during various stages of this project.

Lastly, we are thankful to our **family and friends** for their constant support and motivation, which played a vital role in the completion of this research.

REFERENCES

[1] E. Lagren, ‘‘Artificial intelligence as a tool in social media content moderation,’’ Univ. Jyväskylä, Bachelor’s Thesis, Dept. Inf. Syst., Jyväskylä, Finland, 2023, pp. 1–27.
[2] ALCOFORADO, A. et al. Zeroberto: Leveraging zero-shot text classification by topic modelling. In: PINHEIRO, V. et al. (Ed.) Computational Processing of the Portuguese Language. Cham: Springer International Publishing, 2022. p.125-36. ISBN 978-3- 030-98305-5.
[3] K. Vidhya, V. S. Abhishek, G. Akash, S. A. Karti, and R. Harshini, “AI-Powered content moderation,” AIP Conference Proceedings, vol. 2914, no. 1, p. 050027, Dec. 2023, DOI : 10.1063/5.0187620.
[4] T. Gillespie, “Content moderation, AI, and the question of scale,” Big Data & Society, vol. 7, no. 2, p. 2053951720943234, Jul. 2020, DOI: 10.1177/2053951720943234.
[5] H. Sun and W. Ni, “Design and Application of an AI-Based Text Content Moderation System,” Scientific Programming, vol. 2022, no. 1, p. 2576535, 2022, DOI: 10.1155/2022/2576535.