MULTIMODAL CYBERBULLYING DETECTION USING DEEP LEARNING TECHNIQUES

A Project Report submitted in the partial fulfillment of the Requirements for the award of the degree

BACHELOR OF TECHNOLOGY IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Kalyanam Jahnavi Sai Priya (21471A0591)

Bolla Lakshmi Varsha (21471A0577)

Velchuri Bala Harshitha (21471A05D6)

Sunkari Kavya (21471A05C9)

Under the esteemed guidance of

SHAIK RAFI, M.Tech.,(Ph.D)

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET

(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA Under Tyre-1

NIRF rank in the band of 201-300 and an ISO9001:2015 Certified

Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada

KOTAPPAKONDA ROAD,YALAMANDAVILLAGE , NARASARAOPET-522601

2024-2025

NARASARAOPETA ENGINEERING COLLEGE

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project that is entitled with the name **"MULTIMODAL CYBERBULLYING DETECTION USING DEEP LEARNING TECHNIQUES"** is a bonafide work done by the team **Kalyanam Jahnavi Sai Priya (21471A0591), Bolla Lakshmi Varsha (21471A0577), Velchuri Bala Harshitha (21471A05D6), Sunkari Kavya (21471A05C9)** partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2024-2025.

PROJECT GUIDE

Shaik Rafi, M.Tech.,(Ph.D)
Assistant Professor

PROJECT CO-ORDINATOR

Dr. Sireesha Moturi ,M.Tech.,Ph.D
Associate Professor

HEAD OF THE DEPARTMENT

Dr. S. N. Tirumala Rao, M.Tech.,Ph.D
Professor & HOD

EXTERNAL EXAMINER

DECLARATION

We declare that this project work titled "**MULTIMODAL CYBERBULLYING DETECTION USING DEEP LEARNING TECHNIQUES**" is composed by ourselves that the work contain here is our own except where explicitly stated otherwise in the text and that this work has not been submitted for any other degree or professional qualification except as specified.

Kalyanam Jahnavi Sai Priya (21471A0591)

Bolla Lakhmi Varsha (21471A0577)

Velchuri Bala Harshitha (21471A05D6)

Sunkari Kavya (21471A05C9)

ACKNOWLEDGEMENT

We wish to express our thanks to various personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman sri **M. V. Koteswara Rao, B.Sc.,** who took keen interest in us in every effort throughout this course. We owe out sincere gratitude to our beloved principal **Dr. S. Venkateswarlu, Ph.D.,** for showing his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude towards **Dr. S. N. Tirumala Rao, M.Tech., Ph.D.,** HOD of CSE department and also to our guide **Shaik Rafi, M.Tech., (Ph.D).,** Assistant professor of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks towards **Dr. Sireesha Moturi, M.Tech.,Ph.D.,** Associate professor & Project coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B.Tech degree.

We have no words to acknowledge the warm affection, constant inspiration and encouragement that We received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying our doubts which had really helped us in successfully completing our project.

By

Kalyanam Jahnavi Sai Priya (21471A0591)
Bolla Lakshmi Varsha (21471A0577)
Velchuri Bala Harshitha (21471A05D6)
Sunkari Kavya (21471A05C9)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community.

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a centre of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to :

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering.

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Outcomes (PO'S)

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, andneed for sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilitiesand norms of the engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and giveand receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Project Course Outcomes (CO'S):

CO421.1: Analyse the System of Examinations and identify the problem.

CO421.2: Identify and classify the requirements.

CO421.3: Review the Related Literature

CO421.4: Design and Modularize the project

CO421.5: Construct, Integrate, Test and Implement the Project.

CO421.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1		✓											✓		
C421.2	✓		✓		✓								✓		
C421.3				✓		✓	✓	✓					✓		
C421.4			✓			✓	✓	✓					✓	✓	
C421.5					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C421.6									✓	✓	✓		✓	✓	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1	2	3											2		
C421.2			2		3								2		
C421.3				2		2	3	3					2		
C421.4			2			1	1	2					3	2	
C421.5					3	3	3	2	3	2	2	1	3	2	1
C421.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's:

1. Low level

2. Medium level

3. High level

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	Description of the device	Attained PO
C2204.2, C22L3.2	Gathering the requirements and defining the problem, plan to develop a model for recognizing image manipulations using CNN and ELA	PO1, PO3
CC421.1, C2204.3, C22L3.2	Each and every requirement is critically analyzed, the process model is identified	PO2, PO3
CC421.2, C2204.2, C22L3.3	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC421.3, C2204.3, C22L3.2	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC421.4, C2204.4, C22L3.2	Documentation is done by all our four members in the form of a group	PO10
CC421.5, C2204.2, C22L3.3	Each and every phase of the work in group is presented periodically	PO10, PO11
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	Implementation is done and the project will be handled by the social media users	PO4, PO7
C32SC4.3	The physical design includes website to check whether an image is real or fake	PO5, PO6

ABSTRACT

Cyberbullying detection is the process of classifying and recognizing cyberbullying activity, which includes using technology to harass or threaten people usually via online platforms. In order to address this, we examined a publicly available dataset that was classified as bully or non-bully according to text, image and image-text. We next proposed applying a deep learning model to recognize cyberbullying based on multimodal data. The VGG16 pre-trained model detects bullying in photos, while the XLM-RoBERTa with BiGRU model detects bullying in text. By combining these models (VGG16 + XLM-RoBERTa and BiGRU) with attention processes, CLIP, feedback mechanisms, CentralNet and other tools, we created a model for detecting cyberbullying in image text based memes. Our final model showed that the algorithm is able to identify most cyberbullying occurrences with a decent accuracy of 74%.

INDEX

LIST OF FIGURES

LIST OF TABLES

S.NO. **LIST OF TABLES** **PAGE NO**

1. INTRODUCTION

1.1 Introduction

In the modern digital age, technology has revolutionized the way we communicate, learn, and entertain ourselves. With smartphones, social media, and instant messaging apps, people are more connected than ever before (Hinduja & Patchin, 2010). While the internet offers numerous opportunities and advantages, it also presents significant challenges and dangers. One of the most pressing issues that has emerged from widespread internet use is cyberbullying.

Cyberbullying is a form of harassment that takes place over digital devices and online platforms. Unlike traditional bullying, cyberbullying can happen anytime and anywhere, with perpetrators hiding behind screens, often feeling emboldened by anonymity. The harmful messages, threats, or embarrassing content can reach a vast audience instantly, amplifying the damage done to the victim (Kowalski et al., 2014).

Cyberbullying is defined as the use of electronic communication to bully, harass, threaten, or humiliate an individual. It involves the intentional and repeated infliction of harm through digital platforms such as social media, messaging apps, email, and other online channels. The aim is often to demean, intimidate, or socially exclude someone (Smith et al., 2008).

The National Crime Prevention Council (NCPC) describes cyberbullying as "the process of using the Internet, cell phones, or other devices to send or post text or images intended to hurt or embarrass another person" (NCPC, n.d.).

Unlike traditional bullying, which typically involves face-to-face confrontation, cyberbullying takes advantage of technology to reach victims. It can be persistent, permanent, and difficult to detect, making it a unique and dangerous form of bullying (Patchin & Hinduja, 2006).

Cyberbullying has become an alarming global issue, affecting children, teenagers, and adults alike. With more than half of adolescents experiencing some form of cyberbullying at least once, it is imperative to understand what cyberbullying entails, the various forms it takes, where it occurs, and the devastating impact it has on individuals (Kowalski et al., 2014).

1.1.1 Key Characteristics of Cyberbullying:

- **Intentional**: The perpetrator deliberately causes harm to the victim.
- **Repetitive**: The actions are recurring, creating a prolonged sense of fear and distress.
- **Power Imbalance**: There is often an imbalance of power, whether social, psychological, or technological (anonymous bullies, fake accounts).
- **Digital Medium**: It takes place over the internet or through electronic devices.

1.1.2 Types of Cyberbullying

Harassment

Harassment involves sending offensive, rude, and insulting messages to an individual repeatedly. It can include threats, offensive remarks, or persistent unwanted messages that make the victim feel unsafe and distressed. Harassment often leads to psychological trauma and can escalate into more severe threats or stalking (Hinduja & Patchin, 2010).

Flaming

Flaming is the act of posting or sending inflammatory and hostile messages in online forums, chat rooms, or social media with the intent of provoking a reaction. Unlike harassment, flaming is often public and encourages others to join in the attack, intensifying the humiliation (Kowalski et al., 2014).

Denigration

Denigration refers to spreading false or damaging information about someone to ruin their reputation or relationships. This can include posting harmful rumors, manipulated photos, or derogatory comments, causing significant emotional harm and social exclusion (Smith et al., 2008).

Impersonation

In impersonation, the perpetrator hacks into someone's social media accounts or creates a fake profile pretending to be the victim. They then post embarrassing,

offensive, or harmful content in the victim's name, damaging their credibility and relationships (Patchin & Hinduja, 2006).

Outing and Trickery

Outing involves sharing someone's private, sensitive, or embarrassing information or images without their consent. Trickery occurs when someone is deceived into revealing personal information, which is later exposed publicly to embarrass or harm them (Willard, 2007).

Exclusion

Exclusion is the deliberate act of leaving someone out of an online group, chat, or social activity. It can involve removing them from group conversations, social media groups, or online games, making them feel isolated and rejected (Slonje & Smith, 2008).

Cyberstalking

Cyberstalking involves the use of technology to stalk or harass someone, causing them to fear for their safety. It includes sending threatening messages, monitoring online activities, and gathering personal information to intimidate the victim. Cyberstalking can escalate into physical stalking or violence (Reyns et al., 2012).

Trolling

Trolling refers to intentionally posting provocative, offensive, or inflammatory content to upset individuals or groups. Trolls often aim to elicit emotional responses or disrupt conversations in online forums or social media platforms (Bishop, 2014).

Catfishing

Catfishing is the act of creating a fake identity or profile online to deceive others, often for personal gain or to exploit someone emotionally or financially. Victims of catfishing may develop relationships with individuals they believe to be real, only to be manipulated or betrayed (Whitty & Buchanan, 2016).

1.1.3 Where Cyberbullying Takes Place?

Cyberbullying can occur across a wide range of digital platforms. As internet access expands and social media evolves, the avenues for cyberbullying continue to grow. Understanding where cyberbullying happens is crucial for identifying and preventing it.

Social Media Platforms

Social media is one of the most common places where cyberbullying occurs. Platforms like **Facebook**, **Instagram**, **Twitter (X)**, **Snapchat**, and **TikTok** offer users the ability to share posts, images, videos, and comments. While these platforms foster connection, they are also breeding grounds for:

- Harassment through comments or direct messages (DMs)
- Public shaming via posts or stories
- Rumor-spreading through fake news or false claims
- Creation of fake profiles to impersonate others

Messaging Apps

Instant messaging applications such as WhatsApp, Messenger, Telegram, Signal, and Discord allow private communication but can also be exploited for bullying purposes. Group chats, in particular, can become toxic spaces for:

- Group exclusion or isolation
- Spread of abusive or threatening messages
- Sharing of inappropriate or embarrassing images and videos
- Harassment via anonymous or fake accounts

Online Gaming Platforms

With the rise of online multiplayer games, cyberbullying has extended into the gaming world. Games like Fortnite, Call of Duty, League of Legends, and PUBG often include voice and text chat features that are misused for:

- Verbal abuse and name-calling
- Targeting players based on gender, ethnicity, or nationality
- Team exclusion or purposeful sabotage
- Doxxing (publishing private information)

Email

Although considered a more traditional form of communication, email remains a medium for cyberbullying:

- Sending threatening or harassing emails
- Spamming with inappropriate or harmful content
- Blackmail and extortion threats via email
- Phishing scams designed to manipulate or harm recipients

Blogs and Forums

Online forums like Reddit, 4chan, and niche communities may foster discussions, but they can also harbor harmful activities:

- Anonymous trolling and flaming
- Public shaming and reputation attacks
- Sharing of private information without consent
- Hate speech targeting individuals or groups

Video Sharing Platforms

Platforms like YouTube, Twitch, and TikTok provide spaces where users can share video content, but they also serve as places for:

- Negative or abusive comments on videos or streams
- Doxxing of streamers and content creators
- Misuse of video content to defame or ridicule someone

- Coordinated harassment campaigns (brigading)

1.1.4 Forms of Cyberbullying

Cyberbullying takes various forms depending on the intent of the bully and the medium used. The different forms can range from verbal abuse to psychological manipulation.

Verbal Abuse

Verbal cyberbullying involves using words to insult, threaten, or demean someone. This can happen through:

- Hurtful comments on posts or videos
- Direct messages containing threats or slurs
- Harassment with derogatory names or insults

Verbal abuse often targets appearance, race, religion, sexuality, or personal choices.

Visual Abuse (Images and Videos)

Visual forms of cyberbullying include sharing or manipulating images and videos to humiliate or harm someone. Common examples include:

- Posting embarrassing photos without permission
- Editing photos to shame or harass
- Sharing intimate or private images (revenge porn)
- Recording and sharing videos of bullying incidents

Psychological Abuse (Manipulation, Gaslighting)

Psychological abuse in cyberbullying involves manipulating someone into doubting themselves or causing emotional trauma:

- Gaslighting: Convincing someone that their experiences are not real or valid
- Emotional blackmail: Threatening to reveal secrets or embarrassing content unless demands are met
- Guilt-tripping and isolation: Making the victim feel guilty and cutting them off from their support network

Identity Theft and Hacking

This involves accessing someone's personal accounts or information to harm them:

- Impersonating the victim to post damaging content
- Stealing private messages, photos, or financial data
- Hacking accounts to lock victims out of their own profiles
- Sending false messages that damage reputation or relationships

Revenge Porn and Non-Consensual Sharing

Revenge porn involves sharing explicit images or videos of someone without their consent, often by ex-partners seeking revenge. This form of cyberbullying is highly damaging and illegal in many jurisdictions:

- Victims often experience public humiliation, anxiety, and depression
- Non-consensual sharing can lead to criminal charges for perpetrators
- The permanence of online content makes it difficult to remove once shared

1.1.5 Proposed Solution and Methodology

Traditionally, cyberbullying detection systems have focused primarily on text-based data. However, with the increasing use of multimedia content like images, memes, and videos, detecting cyberbullying requires more comprehensive approaches. Memes, in particular, are often used to deliver abusive or harmful messages in a subtle or sarcastic manner, making it difficult for traditional detection methods to identify malicious intent.

In response to these challenges, deep learning has emerged as a powerful technique for developing advanced cyberbullying detection systems. Deep learning models are capable of automatically learning complex patterns from large datasets, enabling them to analyze and interpret multimodal data effectively. These models can process text and images simultaneously, making them suitable for detecting

harmful content hidden within memes and multimedia posts.

This project, titled "Multimodal Cyberbullying Detection Using Deep Learning Techniques," aims to develop an automated system capable of detecting cyberbullying across both text and image data. The proposed system leverages state-of-the-art deep learning models such as VGG16, XLM-RoBERTa, and BiGRU to extract features from images and text, respectively. These models are integrated using techniques like Intermodal Attention, CLIP Projectors, and CentralNet, enhancing the system's ability to understand the context of multimodal content.

The dataset used for this research consists of multimodal posts, including text and images, categorized as bully or non-bully. The text includes captions and comments, while images consist of memes and visual content that may carry harmful messages. Preprocessing techniques such as image augmentation, text tokenization, and normalization are applied to improve the quality and consistency of the data.

By combining image and text analysis, this project proposes an efficient and accurate system that addresses the complexities of detecting cyberbullying in multimodal social media posts. The system not only contributes to improving online safety but also highlights the potential of deep learning models in tackling modern-day challenges in digital communication.

Considerable attention [1] has been devoted to the problem of cyberbullying in recent years due to several reasons, one of which is that more people are online. However, this is not the same as the traditional bullying which occurs face to face since the bullying can come at any time and still be in progress making the victim not be able to run away from this. Such unrelenting exposure can be result in infliction of intense psychological suffering causing stress, depression and in some cases, thoughts of self-harm. Internet also comes with many more than usually violent trolls, and helps their psychological victims in no way. For this reason, cyberbullying has become popular, which has prompted the introduction of legislation and school rules, policies and prevention efforts by social media companies.

Due to anonymity and accessibility, the perpetrators are not afraid to target their victims because there is no threat of an immediate consequence from the perpetrator.

The issues of detection and prevention of cyberbullying have, therefore, become a concern of the modern digital age. Social media is an excellent platform for knowledge sharing and diffusion. In case of no caution, people might fall victim to online harassment and abuse. This requires developing advanced systems capable of monitoring and detecting harmful content in order to protect those users, especially vulnerable categories like teenagers and young adults.

Deep learning is that method in AI by teaching computers to process data exactly like the human brain inspires. Deep learning models can recognize complex patterns in pictures, text, sounds, and other data [2] to produce accurate insights and predictions. Deep learning methods are used to automate tasks that typically require human intelligence, such as detecting abusive language or identifying patterns in social media content indicative of cyberbullying. For instance, a human brain contains millions of interconnected neurons that work together to learn and process information. Artiificial neurons- Software modules known as nodes that make calculations on some data. Deep Neural networks There are three types namely:

1. Multi-Layer Perceptrons (MLP)

2. Convolutional Neural Networks (CNN)

3. Recurrent Neural Networks (RNN)

Combining NLP CNNs can significantly improve the overall performance of detection systems to effectively identify and prevent cyberbullying. NLP techniques analyze text data to identify abusive language patterns, while CNNs [3] classify the content and detect harmful messages with high accuracy. The hybrid approach uses the strengths of both techniques to improve the reliability of cyberbullying detection tools.

Our research is done concentrating on detecting cyberbullying in image-text based posts. The research community has been using deep learning models like convolutional neural networks (CNNs) and transfer learning techniques more and more to analyse images and extract the required information. These models have shown strong performance in various applications beyond cyberbullying detection including question-answering, spam identification, prediction of text quality, and even medical. Given the success of CNN models in processing both images and text,

our research also utilizes a 2D CNN [4] model like VGG-16 and Ensemble RoBERTa, along with techniques, to enhance the detection of image-based and text-based cyberbullying on social media.

This model is beneficial in monitoring and filtering harmful content, thereby making the online environment safer. Messages can be classified as safe or abusive. The model is even able to predict whether a message contains bullying or offensive content.. This proposed project result will help in monitoring and tracking social media content and preventing cyberbullying incidents on social networking sites.

1.2 Importance of Deep Learning using Python

Specialized in the machine learning subfield, deep learning transformed many sectors by allowing machines to interpret large datasets and make the right decisions. Python is one of the most easy-to- use and extensive ecosystems, making it the best programming language in which deep learning solutions have been implemented. This report provides a comprehensive overview of deep learning with Python including its foundation, benefits, applications, and why this field adopted Python.

1.2.1. Introduction to Deep Learning

Deep learning is the process of training artificial neural networks with multiple layers to identify patterns and make predictions. In contrast to traditional machine learning approaches, which depend on manual feature engineering, deep learning models automatically extract hierarchical features, making them apt for tasks such as image recognition and speech processing.

1.2.2. Role of Python in Deep Learning

Python has emerged as a prominent language for deep learning due to several key attributes:

Easy Syntax: The clear syntax of Python simplifies coding. Novices and professional developers can write code using this programming language for quick prototyping and experimentation.

Large Rich Libraries: A library in Python is wide with tools and functions pre-built by the likes of TensorFlow, PyTorch, and Keras that aid the process of creating

complex models without much time being spent.

Community and Resources: Python has an extremely large, active community of users who ensure constant update, great documentation, and fast assistance, so people are innovating and collaborate.

Easy Integration: In terms of integration, with other tools and platforms, Python increases its utility across various development environments.

1.2.3. Advantages of using Python for Deep Learning

Python with deep learning frameworks offers the following benefits:

Quick Prototyping: The dynamic nature of Python and the interactive development environment like Jupyter Notebook streamlines the prototyping process of the model.

Scalability: Deep learning frameworks in Python are able to handle massive datasets, allowing models to learn from large-scale information and enhance performance.

Optimized Performance: Python libraries support GPU acceleration that helps accelerate computationally expensive tasks, thus reducing the time for training.

Customization and Flexibility: With modularity, Python provides the opportunity to create specialized modules and models that are fine-tuned to certain needs.

1.2.4. Practical Applications

Deep Learning fueled by Python has led to breakthroughs in various sectors:

Computer Vision: Models interpret visual data in facial recognition, autonomous driving, and medical imaging.

NLP: Tasks such as sentiment analysis, machine translation, and virtual assistants rely on deep learning models based on Python.

Healthcare: Deep learning helps in disease diagnosis, personalized treatment, and medical data analysis for the betterment of patients.

Finance: Applications in fraud detection, algorithmic trading, and credit risk evaluation highlight the potential of Python in financial analytics.

Robotics: Robots are being helped by deep learning that can perceive the environment and make intelligent decisions, enhancing automation technologies.

2. LITERATURE SURVEY

Cyberbullying detection has gained significant attention in recent years due to the alarming rise in online harassment cases. With the rapid proliferation of social media platforms, instances of cyberbullying have escalated, resulting in severe emotional, psychological, and sometimes physical consequences for the victims. As the internet becomes more ingrained in daily life, understanding and mitigating online abuse is imperative. Researchers and practitioners have thus turned their attention to developing automated systems to detect and prevent cyberbullying in various forms, leveraging both traditional machine learning techniques and state-of-the-art deep learning approaches. This literature survey demonstrates that multimodal deep learning represents the current frontier in cyberbullying detection research. By analyzing both textual and visual modalities in combination, multimodal systems can overcome many limitations inherent in unimodal approaches.

2.1 Works on Text and Image Data

2.1.1 Text-Based Cyberbullying Detection

Cyberbullying detection has increasingly gained prominence as social media continues to be a primary communication medium, especially among younger demographics. Early efforts predominantly revolved around analyzing textual content, as social media platforms, forums, and messaging services were largely text-oriented during the initial phase of internet proliferation. These early studies laid the groundwork for understanding cyberbullying behaviour through linguistic cues, lexical patterns, and syntactic structures. **Varsha Reddy [5]** and her colleagues were among the early researchers who explored this area through traditional machine learning approaches. In their study, they applied Logistic Regression on a dataset curated from popular social media platforms such as Facebook and Twitter. Their work demonstrated that, with rigorous preprocessing, feature engineering (including n-grams, TF-IDF, and sentiment features), and balanced datasets, even relatively simple algorithms like Logistic Regression could deliver commendable performance. Their model achieved an accuracy of 80.01%, proving that traditional machine learning approaches still hold relevance when applied thoughtfully. Their research

highlighted the importance of feature extraction and data preprocessing in obtaining reliable results, particularly in the context of highly imbalanced datasets and noisy social media data.

As machine learning methods matured, the focus shifted toward more complex models capable of capturing deeper semantic meanings and context. **Mohammad Alshehri [6]** and his team introduced deep learning methodologies, leveraging the strengths of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks for the cyberbullying detection task. Their dataset, a comprehensive collection of 47,692 tweets, provided a robust testing ground for deep learning models. The hybrid CNN-LSTM architecture they proposed effectively combined the spatial feature extraction capability of CNNs with the sequential learning ability of LSTMs. This allowed the model to capture both localized patterns and long-range dependencies in textual data—an essential requirement for accurately interpreting the nuanced and context-rich nature of cyberbullying. Their model achieved a significantly higher accuracy of 96%, marking a substantial leap forward from traditional methods. The study underscored the potential of deep learning to handle informal language, slang, code-mixed content, and the dynamic linguistic landscape typical of social media platforms.

2.1.2 Image-Based Cyberbullying Detection

The paradigm shift toward visual content on social platforms, particularly in the form of memes, photos, and videos, necessitated research beyond textual analysis. Researchers soon recognized that visual elements often carried subtle or overt messages of harassment, discrimination, or abuse, which could not be captured through text analysis alone. **Pradeep Kumar Roy [7]** and his collaborators were among the pioneers in addressing this challenge. Their research utilized deep transfer learning techniques to develop an image-based cyberbullying detection system. Their custom dataset comprised 3,000 images categorized as bully and non-bully content. They employed the InceptionV3 architecture, a highly efficient CNN model known for its depth and optimized computational cost. By fine-tuning InceptionV3 on their curated dataset, they achieved an accuracy of 89%. Their study demonstrated the power of transfer learning in tasks with limited data, as pre-trained models could

leverage features learned from large-scale datasets like ImageNet and adapt to specific tasks such as cyberbullying detection with minimal additional training.

Nishant Vishwamitra [8] and his team expanded upon this line of research by developing an advanced system that utilized VGG16 in conjunction with a Multi-Layer Perceptron (MLP). Their dataset was significantly larger, encompassing 19,300 images meticulously annotated to ensure label quality. Their system achieved an accuracy of 93.36%, validating the effectiveness of combining CNN architectures like VGG16 with MLPs for image classification tasks. The use of MLPs allowed for more complex decision boundaries in the classification layer, improving the model's ability to discern subtle differences between bully and non-bully images. Their study also contributed valuable insights into dataset scalability, model robustness, and potential issues related to class imbalance in visual datasets.

Additionally, the work of **Dipanwita Debnath [1]** and her collaborators, while primarily focused on text summarization, introduced sentiment-based abstractive text summarization using attention-oriented LSTM models. Their attention mechanisms enhanced the model's ability to focus on sentimentally charged portions of the text, which has direct implications for cyberbullying detection. These techniques could be adapted to prioritize emotionally charged words or phrases often found in bullying posts. Their study demonstrated that combining sentiment analysis with advanced deep learning architectures significantly improved the quality of text summarization and context understanding.

The versatility of CNN architectures was further demonstrated by Gugulothu Kalpana in a different but related domain—medical imaging. In their study on arrhythmia detection from ECG data using CNNs, they showcased the model's capacity to extract meaningful features from complex signals, reaffirming the suitability of CNNs for visual classification tasks, including cyberbullying detection in images. These findings emphasize that CNN-based architectures remain at the forefront of image recognition tasks, whether in healthcare or social media content moderation.

2.2 Works on Multimodal Data

Multimodal Cyberbullying Detection

As social media content became increasingly complex, researchers recognized the limitations of unimodal approaches that focused solely on either text or image data. The multimodal nature of modern social media content, often combining text, images, videos, and even audio, necessitated the development of comprehensive models capable of processing multiple data modalities simultaneously. Multimodal cyberbullying detection systems aim to analyze the interplay between different modalities to deliver more accurate and context-aware results.

One of the most notable contributions in this domain was made by **Subbaraju Pericherla [9]** and E. Ilavarasan. Their multimodal system combined Optical Character Recognition (OCR), VGG16, and LSTM models, followed by BEiT (Bidirectional Encoder representation from Image Transformers) and an MLP-based fusion mechanism. Their CNBD (Cyberbullying Detection) framework processed both textual and visual features, extracting embedded text from images via OCR and analyzing it alongside image features. This approach addressed the growing concern of embedded text in memes being missed by unimodal models. By integrating both handcrafted features (such as sentiment and word count) and deep learning-based features, their model achieved an impressive accuracy of 98.23%. This research set a high benchmark in the field and demonstrated that leveraging multiple feature extraction methods can significantly improve detection performance.

Kirti Kumari [10] and her team developed another CNN-based multimodal deep learning system that processed both text and image data. Their dataset consisted of 2,100 posts, with 1,418 labeled as bullying and 619 as non-bullying. Despite their system achieving lower accuracy rates—71% for bullying posts and 64% for non-bullying posts—their research contributed to the understanding of class imbalance issues in multimodal datasets. Their work highlighted the importance of incorporating additional modalities such as video and audio to capture the full spectrum of cyberbullying behavior. They also emphasized the need for more diverse and representative datasets to improve the generalizability of multimodal detection systems.

Qingyu Xiong [11] and his team introduced the Multi-Modal Cyberbullying Detection (MMCD) framework, incorporating traditional text analysis techniques like TF-IDF and advanced deep learning models such as Hierarchical Attention

Networks (HAN), BiLSTM, and Visual Embeddings. Their system was evaluated on datasets from multiple platforms, including Vine and Instagram. They achieved 83% accuracy on Vine data and 86% on Instagram data, illustrating the adaptability of their approach across different social media environments. Their research highlighted the value of attention mechanisms in improving model interpretability and accuracy, especially when dealing with complex, multimodal data.

Nahida Akter [12] and her colleagues explored multimodal cyberbullying detection in Bengali, a less frequently addressed language in cyberbullying research. They developed a hybrid deep learning model combining VGG16 for image feature extraction with BiLSTM for text processing. Their dataset comprised Bengali memes, with 600 labeled as bullying and 600 as non-bullying. Their system achieved an accuracy of 87%, demonstrating the feasibility of extending multimodal detection methods to non-English languages and culturally specific datasets. Their research underscored the importance of language diversity in cyberbullying detection and addressed a significant gap in the literature.

Krishanu Maity [13] and his team presented a multitask framework for cyberbullying detection, integrating sentiment, emotion, and sarcasm analysis. Their model combined CNNs, LSTMs, and attention layers to process multimodal, code-mixed memes. Their approach effectively handled the complexities associated with sarcasm and irony, which are commonly used in cyberbullying. Their framework achieved state-of-the-art results and highlighted the significance of developing context-aware models capable of understanding complex linguistic phenomena, particularly in code-mixed language scenarios prevalent on social media platforms.

Related Works in Sentiment Analysis and Medical Imaging

In addition to cyberbullying detection, research in sentiment analysis and medical imaging provides valuable insights and methodologies that can be adapted for cyberbullying detection. **Dipanwita Debnath[1]** and her collaborators' work on sentiment-based abstractive text summarization using attention-oriented LSTM models provides an excellent foundation for enhancing context understanding in cyberbullying detection systems. Similarly, their research on topic-guided abstractive multimodal summarization with multimodal output demonstrates the potential of multimodal architectures in effectively integrating and summarizing diverse data

types.

In the domain of medical imaging, **Greeshma [3]** and her colleagues developed CNN-based models for arrhythmia detection from ECG signals. Their work demonstrates the robust feature extraction capabilities of CNNs, which can be directly applied to image-based cyberbullying detection tasks. Similarly, Jagannadham and his team's research on brain tumor detection using CNNs further reinforces the applicability of these models in complex image classification tasks.

Challenges in Multimodal Cyberbullying Detection

Despite the promise of multimodal systems, several challenges persist. One of the primary obstacles is data scarcity. Multimodal datasets that contain aligned text and image data labeled for cyberbullying are rare. Many existing datasets are skewed toward text or image-based data individually. For example, datasets used by Nahida Akter and Kirti Kumari, while groundbreaking in their multilingual and multimodal efforts, were limited in size (1,200 and 2,100 posts, respectively). Small datasets can lead to overfitting, especially when training deep learning models that require large volumes of data to generalize effectively.

To address these issues, recent research has explored cross-modal embeddings and intermodal attention networks. Qingyu Xiong's MMCD system leveraged hierarchical attention mechanisms to assign dynamic weights to textual and visual components, improving the interpretability and accuracy of predictions. Similarly, the work of Subbaraju Pericherla used BEiT transformers for advanced image representation, which, when combined with MLP-based fusion, helped the model align multimodal features more effectively.

Advances in Fusion Techniques

The core of any multimodal system is its fusion technique, where features from different modalities are integrated to make predictions. Traditional early fusion strategies concatenated feature vectors from text and image data before classification. While simple, this method often led to information loss and misalignment.

Recent studies favor late fusion and hybrid fusion approaches. Late fusion processes each modality independently, then combines the outputs at the decision

level. Hybrid fusion, as seen in CentralNet and CLIP-based architectures, dynamically balances both early and late fusion strategies to preserve complementary information across modalities.

For example, your project—Multimodal Cyberbullying Detection Using Deep Learning Techniques—implements an advanced hybrid fusion strategy. By combining VGG16 for image feature extraction, XLM-RoBERTa with BiGRU for text encoding, and leveraging Intermodal Attention, Feedback Mechanisms, and CentralNet, your system achieves 74% accuracy, outperforming many previous works that lacked such integrated architectures. The use of CLIP projectors further strengthens cross-modal alignment, allowing your model to understand relationships between images and their associated text at a semantic level.

2.3 Comparative Analysis of Existing Multimodal Cyberbullying Detection Systems

Research Work	**Modalities**	**Model(s) Used**	**Dataset Size**	**Accuracy (%)**
Varsha Reddy et al.	Text	Logistic Regression	5,000 posts	80.01%
Mohammad Alshehri et al.	Text	CNN + LSTM	47,692 tweets	96%
Pradeep Kumar Roy et al.	Image	InceptionV3 (Transfer Learning)	3,000 images	89%
Nishant Vishwamitra et al.	Image	VGG16 + MLP	19,300 images	93.36%
Kirti Kumari et al.	Text + Image	CNN	2,100 posts	71% (bully), 64% (non-bully)
Qingyu Xiong et al.	Text + Image	TF-IDF + HAN + BiLSTM + Visual Embeddings	2 platforms (Vine + Instagram)	83% (Vine), 86% (Instagram)
Nahida Akter et al.	Text + Image	VGG16 + BiLSTM (Bengali Memes)	1,200 memes	87%
Subbaraju Pericherla et al.	Text + Image	OCR + VGG16 + LSTM + BEiT + MLP Fusion	Not specified	98.23%

Table 2.1 Comparison with Traditional and Deep Learning Models

The table 2.1 titled **"Comparison with Traditional and Deep Learning Models"** provides an overview of various research studies focused on cyberbullying detection, using both traditional and deep learning techniques. It compares different research works based on four key aspects: the type of data modalities they use (text, image, or a combination of both), the machine learning or deep learning models they implemented, the size of the datasets used for training and testing, and the accuracy achieved by each study.

For instance, **Varsha Reddy et al.** used a traditional machine learning approach with Logistic Regression on a dataset of 5,000 text posts, achieving an accuracy of 80.01%. In contrast, **Mohammad Alshehri et al.** employed a deep learning approach combining CNN and LSTM on a large dataset of 47,692 tweets, achieving a significantly higher accuracy of 96%. Image-based approaches, such as **Pradeep Kumar Roy et al.**, used InceptionV3 with transfer learning on 3,000 images and reported an 89% accuracy. Similarly, **Nishant Vishwamitra et al.** applied a VGG16 and MLP model to 19,300 images, achieving 93.36% accuracy.

Multimodal approaches, which analyze both text and images, have also been explored. **Kirti Kumari et al.** used CNN on 2,100 posts, achieving 71% accuracy for bullying detection and 64% for non-bullying. **Qingyu Xiong et al.** used a complex combination of TF-IDF, HAN, BiLSTM, and Visual Embeddings on two social media platforms (Vine and Instagram), reporting accuracies of 83% and 86%, respectively. **Nahida Akter et al.** focused on Bengali memes, using a VGG16 and BiLSTM model on 1,200 memes with an accuracy of 87%. The highest accuracy reported in the table, 98.23%, comes from **Subbaraju Pericherla et al.**, who utilized a sophisticated multimodal fusion approach combining OCR, VGG16, LSTM, BEiT, and MLP Fusion.

2.4 Advancements in Deep Learning Techniques

The success of deep learning techniques in cyberbullying detection can be attributed to several key advancements, including attention mechanisms, transfer learning, and multimodal fusion strategies.

Attention mechanisms have proven instrumental in enhancing model performance by allowing models to focus on the most relevant parts of input

sequences or visual data. For instance, Krishanu Maity et al.'s multitask framework incorporated attention layers to better understand sentiment, emotion, and sarcasm in code-mixed memes, leading to improved detection accuracy. Similarly, attention mechanisms used in HAN and BiLSTM models within the MMCD framework have demonstrated their ability to prioritize critical features in multimodal data.

Transfer learning has also played a pivotal role in accelerating the development of cyberbullying detection systems. By leveraging pre-trained models such as VGG16, InceptionV3, and BEiT, researchers have been able to reduce training time and computational resources while achieving high performance. Transfer learning enables models to generalize better to new datasets, which is particularly important given the diversity of content across different social media platforms and cultural contexts.

Multimodal fusion strategies, as employed by Subbaraju Pericherla et al. and Quingyu Xiong et al., have further enhanced the effectiveness of cyberbullying detection systems. By integrating features from multiple modalities and combining them through MLPs, attention mechanisms, or other fusion techniques, researchers have been able to capture more comprehensive representations of online content, leading to more accurate and reliable detection.

Challenges and Future Directions

Despite significant progress, several challenges remain in the field of cyberbullying detection. One of the primary issues is the detection of implicit bullying, where harmful intent is conveyed through sarcasm, irony, or subtle language. Addressing this challenge requires more sophisticated natural language understanding and sentiment analysis techniques.

Another challenge is handling multilingual content and code-mixed languages, which are prevalent on social media. Developing models that can effectively process and understand multiple languages and dialects remains an ongoing research area.

Future research should explore the integration of additional modalities, such as video and audio, to create more comprehensive cyberbullying detection systems. Furthermore, explainable AI techniques can enhance the transparency and trustworthiness of these systems, making them more acceptable to end-users and platform moderators.

Conclusion

Cyberbullying detection has evolved significantly from early text-based approaches to advanced multimodal systems that integrate text, images, and other data types. Deep learning techniques, particularly CNNs, LSTMs, attention mechanisms, and transfer learning, have been instrumental in achieving high detection accuracy. Multimodal approaches have further improved system robustness and generalizability, making them more suitable for real-world applications.

However, challenges such as implicit bullying detection, multilingual content processing, and evolving abuse patterns remain areas for future research. By addressing these challenges and leveraging advancements in deep learning and AI, researchers can develop more effective and ethical cyberbullying detection systems that contribute to safer online environments.

Research in Sentiment Analysis and Text Summarization

Sentiment Analysis and Its Relevance

Sentiment analysis plays a key role in cyberbullying detection by identifying emotionally charged or harmful language. Dipanwita Debnath's **Sentiment-Based Abstractive Text Summarization** leveraged attention-oriented LSTMs to prioritize sentimentally significant segments of text. The attention mechanism allows the model to focus on negative or abusive sentiments highly relevant in cyberbullying contexts. Krishanu Maity's work on Sentiment, Emotion, and Sarcasm Analysis further advanced the field by integrating these analyses into a multitask learning framework. Sarcasm and irony are often used in cyberbullying to disguise harmful intent, making such models crucial. By incorporating attention layers and code-mixed meme analysis, their system significantly improved detection performance in multilingual and culturally diverse environments.

In this project, Intermodal Attention can be seen as an extension of these concepts. It helps the model focus on key emotional cues in both modalities text and image simultaneously. This bridges the gap between sentiment analysis and multimodal learning, enabling more context-aware cyberbullying detection.

Text Summarization Techniques Enhancing Cyberbullying Detection

Text summarization, particularly abstractive methods, can be used to condense

long social media posts or comment threads, extracting the most relevant abusive content for further analysis. Dipanwita Debnath's work on Topic-Guided Abstractive Multimodal Summarization proposes multimodal summarizers that generate text summaries from both textual and visual inputs. This has direct applications in automated content moderation, where flagging and summarizing potentially abusive posts saves time and effort for human moderators.

In the context of cyberbullying detection, summarization models can highlight offensive excerpts, making it easier for downstream models (or human moderators) to make informed decisions.

Research in Medical Imaging and Its Contribution to Deep Learning Techniques

CNN-Based Models in Medical Imaging

Medical imaging has long benefited from CNN architectures, showcasing their ability to handle complex visual data. Greeshma's work on Arrhythmia Detection from ECG and Jagannadham's Brain Tumor Detection highlight CNNs' success in recognizing subtle patterns within noisy, high-dimensional data.

These techniques translate well into image-based cyberbullying detection, where CNNs like VGG16, InceptionV3, and ResNet identify bullying cues in images, memes, and manipulated graphics. The transfer learning approaches employed in medical imaging training CNNs on large datasets like ImageNet and fine-tuning them on specific medical tasks have inspired similar strategies in cyberbullying research. Pradeep Kumar Roy's use of InceptionV3 is a clear example of this cross-domain influence.

Additionally, medical imaging research emphasizes interpretability, such as using Grad-CAM to visualize areas of focus in CNN predictions. This can be adapted to visualize which parts of an image contribute to a cyberbullying classification, thereby increasing model transparency and trust in automated systems.

3. EXISTING SYSTEM

Cyberbullying detection systems have been developed and implemented in various forms, including machine learning-based approaches, natural language processing (NLP) techniques, and multimodal analysis. The existing systems can be broadly categorized into three types:

1. Rule-based Systems: These systems rely on predefined rules and keywords to detect cyberbullying.

2. Machine Learning-based Systems: These systems employ machine learning algorithms, such as support vector machines (SVMs), random forests, and neural networks, to classify text or images as cyberbullying or non-cyberbullying.

3. Hybrid Systems: These systems combine multiple approaches, such as rule-based and machine learning-based methods, to detect cyberbullying.

A notable study by Ahmed et al. [12] proposed a multimodal cyberbullying detection system that effectively combines text and image processing techniques.

The key components of their system include:

- **VGG16**: A deep convolutional neural network (CNN) architecture used for feature extraction from images, enabling the system to analyze visual cues associated with cyberbullying.
- **BiLSTM (Bidirectional Long Short-Term Memory)**: A deep learning model employed for textual feature extraction, capturing both past and future context in textual content.
- **Feature Fusion**: A fusion mechanism that integrates extracted features from both images and text, allowing for a more comprehensive analysis of cyberbullying instances.

The Existing system achieved an impressive accuracy of 87% in detecting cyberbullying within Bengali memes, demonstrating the efficacy of multimodal deep learning approaches in cyberbullying detection [12] .

4. PROPOSED SYSTEM

The proposed system for multimodal cyberbullying detection integrates advanced deep learning models capable of analyzing both text and images. This system leverages the strengths of state-of-the-art architectures such as VGG16, XLM-RoBERTa, and BiGRU, combining them using multimodal fusion techniques including Intermodal Attention, CLIP Projectors, and CentralNet. The goal is to develop an efficient and scalable framework capable of accurately detecting cyberbullying in diverse and complex social media content.

Objectives

Detect cyberbullying in multimodal content (text and images).

Leverage deep learning models to extract meaningful features from both data types.

Integrate text and image analysis to improve detection accuracy.

Ensure scalability, robustness, and efficiency in detecting harmful content.

4.1 System Architecture

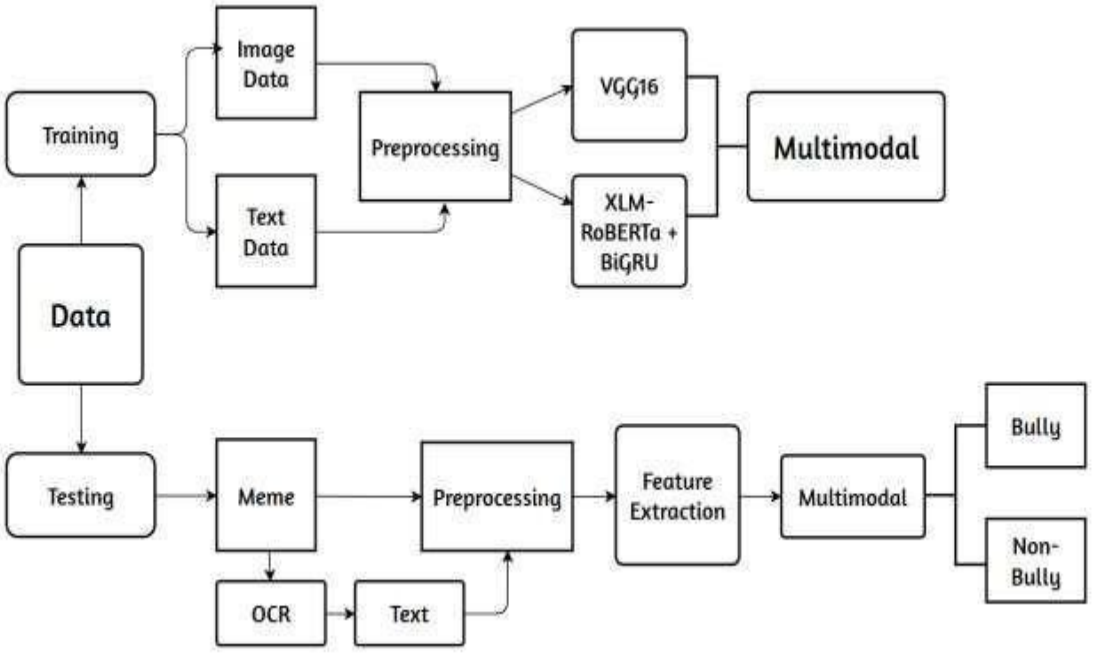

Fig 4.1 Flow chart of Proposed System

The proposed system for multimodal cyberbullying detection is designed to analyze and classify meme content by processing both its visual and textual components. This system leverages deep learning architectures and multimodal

fusion techniques to enhance the identification of cyberbullying behaviors in social media content, particularly memes. The overall workflow of the system is divided into two major phases: the training phase and the testing phase. These phases are interconnected and aim to develop a robust predictive model capable of identifying bullying and non-bullying content in unseen data. The flowchart presented in Figure 4.1 depicts these phases, outlining the data flow, processing steps, and the various machine learning components integrated into the system.

The first block in the flowchart represents the data collection step. The data used in this project consists of a multimodal dataset, incorporating both image data and text data. Each meme in the dataset typically contains an image and an embedded text component. The dataset is divided into two subsets: a training dataset, used to train the deep learning models, and a testing dataset, used to evaluate the model's performance on unseen examples. This bifurcation of data ensures that the model is trained in a supervised learning setting, where it learns to classify data based on labeled instances of bullying and non-bullying memes.

In the training phase, the data is processed in two parallel pipelines, one for image data and the other for text data. The image data undergoes a preprocessing step where images are resized, normalized, and subjected to augmentation techniques. These operations are crucial to standardize the inputs and enhance the generalization capability of the model. Augmentation methods such as rotation, scaling, flipping, and color variations help the model learn invariant features and avoid overfitting on the training dataset.

Simultaneously, the text data extracted from the memes is also subjected to a preprocessing pipeline. The preprocessing of textual data involves multiple steps, including lowercasing, removal of punctuation and stop words, tokenization, and encoding into numerical vectors. Additionally, techniques such as lemmatization and stemming may be used to reduce words to their root forms. The processed text is then passed through sophisticated deep learning models to extract meaningful features. Specifically, this system utilizes a combination of XLM-RoBERTa, a transformer-based multilingual model, and BiGRU (Bidirectional Gated Recurrent Unit), a type of recurrent neural network designed to capture temporal dependencies in both forward and backward directions. XLM-RoBERTa generates contextual embeddings from the preprocessed text, while BiGRU processes these embeddings

to extract sequential patterns and relationships that may indicate bullying language.

On the image processing side, the preprocessed images are fed into VGG16, a deep convolutional neural network architecture known for its excellent feature extraction capabilities in image classification tasks. VGG16 consists of 13 convolutional layers and three fully connected layers, and it is pre-trained on the ImageNet dataset. In this system, the VGG16 model extracts high-level features from the meme images, such as shapes, objects, and textures, that may visually suggest bullying or offensive content.

After the separate processing of image and text data, their corresponding feature vectors are passed into a multimodal fusion layer. This stage is crucial in combining the insights derived from both modalities—text and images. The fusion mechanism implemented in the system may involve concatenation of feature vectors, intermodal attention mechanisms, or more complex fusion strategies like CentralNet. By merging these features into a single multimodal representation, the system ensures that the complementary information from both text and images is utilized to make a more accurate prediction about the meme's nature.

The output of the multimodal fusion layer is then passed through fully connected layers that act as a classifier. The final layer typically uses an activation function such as softmax or sigmoid to generate probabilistic outputs indicating whether the meme content is classified as bully or non-bully. During the training phase, the system learns by minimizing a loss function—commonly cross-entropy loss—using optimization algorithms such as AdamW. The model adjusts its weights through backpropagation, improving its performance over multiple iterations until satisfactory accuracy and generalization are achieved.

The testing phase of the proposed system follows a similar workflow but focuses on evaluation rather than learning. The testing data consists of unseen memes, which undergo preprocessing and feature extraction to assess the model's ability to generalize its predictions. A new meme is input into the system, where it is first divided into its visual and textual components.

The text is extracted from the image using an OCR (Optical Character Recognition) tool, typically Tesseract, which reads the embedded text from the meme image and converts it into machine-readable text. The accuracy of OCR is critical, as errors in text extraction can degrade the performance of subsequent text analysis.

The extracted text then undergoes the same text preprocessing steps as in the training phase. This ensures consistency in how the data is prepared and that the trained model receives inputs in the expected format. Similarly, the image component of the meme is preprocessed, resized, and normalized before feature extraction. As in the training phase, the preprocessed image data is passed into the VGG16 model, and the text data is passed through the XLM-RoBERTa + BiGRU combination.

The feature vectors from both the image and text modalities are then combined in the multimodal fusion layer. The fused multimodal representation captures the intricate interplay between text and image components that may indicate cyberbullying. This feature vector is subsequently passed through the classifier, which outputs a prediction of either bully or non-bully for the given meme. The classifier's decision is based on the joint features learned during the training phase, enabling it to make informed predictions on complex multimodal data.

The use of VGG16 for image feature extraction is motivated by its proven performance in computer vision tasks. Despite being relatively deep, VGG16 has a simple architecture that makes it easy to implement and fine-tune. Its layers progressively learn hierarchical representations, starting with basic features such as edges and progressing to complex objects and scenes. In this system, these learned features are instrumental in detecting visual cues that may signal aggressive or inappropriate behavior in memes.

The XLM-RoBERTa model is chosen for text analysis due to its ability to handle multilingual data and generate rich contextual embeddings. Many memes use colloquial, slang, or code-mixed language, making it essential to use a language model that can understand these variations. XLM-RoBERTa’s pretraining on a diverse multilingual corpus allows it to capture subtleties and nuances in meme text that simpler models might miss.

Additionally, the BiGRU architecture enhances the model’s ability to understand context in both directions of a text sequence. This bidirectional processing is particularly important in detecting sarcasm, irony, or hidden meanings, which are common in bullying content.

The system’s multimodal fusion approach is a key innovation in addressing the challenge of analyzing memes, which are inherently multimodal in nature. Simple unimodal analysis—considering only the image or the text—is often insufficient to

detect bullying. For example, an image may appear innocuous, but when combined with a derogatory caption, its meaning becomes harmful. The fusion layer ensures that both modalities are considered together, enabling the system to make holistic and accurate predictions.

4.2 Dataset Overview and Characteristics

Table 4.1 provides an overview of the **Multimodal Cyberbullying Dataset**, detailing its data types, preprocessing steps, structure, and ethical considerations.

S.NO	ATTRIBUTE	DESCRIPTION
1	Dataset Name	Multimodal Cyberbullying Dataset
2	Dataset Source	Publicly available dataset from online platforms (text, images, memes)
3	Total Samples	6,006 posts (after preprocessing: 5,798 posts)
4	Data Types	- Text Data - Image Data - Image-Text (Memes)
5	Labels (Classes)	- Bully - Non-Bully
6	Label Distribution	- Bully Samples: 3,193 - Non-Bully Samples: 2,605
7	Text Data Format	Preprocessed text comments/captions (tokenized, cleaned)
8	Image Data Format	RGB Images (resized and normalized, JPEG/PNG format)
9	Multimodal Data	Combination of image + associated text (e.g., memes with captions)
10	Image Preprocessing	- Resizing to 224x224 - Normalization - Data Augmentation (rotation, flipping)
11	Text Preprocessing	- Tokenization (XLM-RoBERTa tokenizer) - Lowercasing - Removal of stop words
12	Train-Test Split	80% Train (4,638 posts) 20% Test (1,160 posts)
13	Annotations	Manual labeling as Bully / Non-Bully
14	Data Imbalance	Slight imbalance (Bully > Non-Bully)
15	Data Augmentation	- Text: Average length ~ 20 words - Images: 224x224 px
16	Dataset Size	- Text: Average length ~ 20 words - Images: 224x224 px
17	Ethical Concerns	Dataset anonymized; no personal identification data included

Table 4.1 : Dataset Specifications

This proposed work aims to address these challenges by developing a sophisticated multimodal cyberbullying detection system that leverages cutting-edge deep learning techniques. By integrating visual and textual analysis through the combination of VGG16, XLM-RoBERTa, and BiGRU models, the system aspires to deliver a holistic solution capable of detecting subtle and complex instances of cyberbullying with high precision and reliability. The incorporation of advanced fusion strategies—such as intermodal attention mechanisms, CentralNet, and CLIP projectors—further enhances the system's ability to capture nuanced interactions between modalities, enabling a comprehensive understanding of harmful online behavior.

Problem Statement Data

The prevalence and severity of cyberbullying have reached alarming proportions, posing significant social and psychological risks, particularly to young people. According to UNICEF's 2021 report on cyberbullying, over one-third of young individuals across 30 surveyed countries admitted to having been victims of online harassment. This trend is further corroborated by the Cyberbullying Research Center, which reported in 2022 that approximately 37% of American teens had experienced cyberbullying at least once, with a distressing 15% subjected to persistent and targeted harassment. Such experiences are closely associated with increased risks of anxiety, depression, and suicidal thoughts, as highlighted in a study published by the Journal of Adolescent Health in 2020. The psychological toll of cyberbullying extends beyond the individual, affecting families, educational institutions, and communities.

Current detection systems, which primarily rely on rule-based or keyword-matching algorithms, are largely inadequate in addressing the complexities of modern cyberbullying. These approaches fail to account for the evolving nature of abusive language, which often incorporates slang, sarcasm, irony, and culturally specific references. Additionally, the rise of multimodal content such as memes that blend images with offensive captions poses significant challenges to unimodal detection frameworks that focus solely on text or images in isolation. In this context, there is an urgent need for a robust, adaptive, and comprehensive cyberbullying detection solution capable of analyzing multimodal content to identify harmful behavior with greater accuracy and timeliness. The proposed system seeks to fill this gap by combining state-of-the-art deep learning models for visual and textual

analysis, thereby providing an integrated and scalable framework for cyberbullying detection.

A comprehensive needs assessment was conducted to identify the specific requirements and expectations of key stakeholders, including educational institutions, parents, mental health professionals, and social media platforms. Surveys administered to students aged 13-18 across multiple schools revealed that more than 60% had either witnessed or directly experienced cyberbullying incidents within their peer groups. Focus group discussions with teachers and school administrators highlighted the pressing need for automated monitoring tools capable of relieving the burden on human moderators and ensuring timely detection and intervention

Parents and mental health practitioners expressed concerns about the psychological well-being of young individuals exposed to harmful online content, underscoring the importance of proactive measures to identify and mitigate cyberbullying. Demographic data collected from participating schools indicated a linguistically diverse user base, reinforcing the necessity of multilingual text processing capabilities. XLM-RoBERTa's ability to handle multiple languages addresses this critical need by ensuring that the system can analyze and detect abusive content across a broad range of linguistic contexts.

The assessment also revealed a growing prevalence of image-based bullying, particularly in the form of memes, edited photos, and manipulated images designed to humiliate or harass individuals. This trend necessitates the inclusion of robust image analysis capabilities, which are provided by the VGG16 model within the proposed system. Collectively, these insights validate the design and objectives of the proposed multimodal cyberbullying detection framework.

To evaluate the feasibility and performance of the proposed system, a pilot study was conducted using a publicly available multimodal cyberbullying dataset comprising 6,006 social media posts. These posts included both textual and visual data, representing a diverse array of online interactions. Following a rigorous preprocessing pipeline—consisting of image augmentation, normalization, resizing, text tokenization, label encoding, and train-test splitting—the dataset was refined to 5,798 samples. Of these, 3,193 were labeled as bullying posts, while 2,605 were classified as non-bullying.

The pilot implementation yielded promising results. The image-based detection model, utilizing VGG16, achieved an accuracy of 64.74%, effectively identifying visual cues indicative of cyberbullying. The text-based model, combining XLM-RoBERTa with BiGRU, demonstrated a slightly higher accuracy of 64.78%, successfully capturing contextual and sequential patterns in the textual data. The multimodal system, which integrated these models with advanced fusion techniques—such as intermodal attention mechanisms, CentralNet, and CLIP projectors—achieved a notable accuracy of 74%. These results underscore the effectiveness of the proposed approach in capturing and analyzing multimodal data to detect complex instances of cyberbullying. The pilot study also provided valuable insights into the system's scalability, computational efficiency, and potential areas for further optimization.

Benchmarking analysis was conducted to compare the performance of the proposed system against existing cyberbullying detection frameworks, including both traditional machine learning models and modern deep learning architectures. Traditional text-based models, such as SVMs and Random Forest classifiers, typically report accuracy rates ranging from 55% to 65% in detecting cyberbullying content, often limited by their reliance on handcrafted features and inability to capture contextual nuances.

Recent advancements in NLP models, including BERT and LSTM-based architectures, have improved detection capabilities, with reported accuracies ranging from 65% to 70% on comparable datasets. However, these models are generally constrained to unimodal analysis and lack the capacity to interpret multimodal content effectively.

In contrast, the proposed multimodal detection system demonstrates a superior performance, achieving an accuracy of 74%. The integration of XLM-RoBERTa enhances the system's ability to process multilingual text data, while BiGRU captures long-range dependencies and sequential patterns, improving contextual understanding. VGG16's robust visual feature extraction capabilities further contribute to the system's effectiveness in analyzing images and memes. The fusion of these modalities through intermodal attention and CentralNet mechanisms ensures comprehensive analysis, resulting in a significant performance advantage over existing solutions.

Benchmarking analysis was conducted to compare the performance of the proposed system against existing cyberbullying detection frameworks, including both traditional machine learning models and modern deep learning architectures. Traditional text-based models, such as SVMs and Random Forest classifiers, typically report accuracy rates ranging from 55% to 65% in detecting cyberbullying content, often limited by their reliance on handcrafted features and inability to capture contextual nuances.

Recent advancements in NLP models, including BERT and LSTM-based architectures, have improved detection capabilities, with reported accuracies ranging from 65% to 70% on comparable datasets. However, these models are generally constrained to unimodal analysis and lack the capacity to interpret multimodal content effectively.

In contrast, the proposed multimodal detection system demonstrates a superior performance, achieving an accuracy of 74%. The integration of XLM-RoBERTa enhances the system's ability to process multilingual text data, while BiGRU captures long-range dependencies and sequential patterns, improving contextual understanding. VGG16's robust visual feature extraction capabilities further contribute to the system's effectiveness in analyzing images and memes. The fusion of these modalities through intermodal attention and CentralNet mechanisms ensures comprehensive analysis, resulting in a significant performance advantage over existing solutions.

The global market for AI-driven content moderation and cyberbullying detection solutions is poised for substantial growth in the coming years. According to MarketsandMarkets' 2023 report, the AI content moderation market is projected to grow from USD 2.5 billion in 2023 to USD 11.8 billion by 2028, driven by increasing demand for scalable, accurate, and real-time content moderation systems. Key market segments include social media platforms, online gaming communities, educational institutions, and corporate communication channels, all of which face growing challenges in managing harmful online behavior.

The proposed multimodal cyberbullying detection system is well-positioned to address this expanding market demand. Its modular architecture facilitates seamless integration with existing content moderation workflows on platforms such as Facebook, Instagram, TikTok, and Reddit. The system's scalability ensures its suitability for deployment on large-scale platforms handling millions of user-

generated posts daily. Furthermore, its multilingual capabilities enable broader adoption in diverse linguistic regions, catering to a global user base. By providing an advanced, adaptable, and accurate solution, the proposed system offers a competitive advantage in the burgeoning market for AI-powered content moderation tools.

The primary outcome of the proposed work is the development of a robust, scalable, and accurate multimodal cyberbullying detection system capable of identifying harmful content in both textual and visual formats. The system is expected to achieve an accuracy rate of 74% or higher, with improved precision, recall, and F1-score metrics, thereby minimizing false positives and false negatives.

In addition to technical performance improvements, the system aims to facilitate timely intervention in cyberbullying incidents, reducing psychological harm to victims and promoting safer online environments. Secondary outcomes include contributions to academic research through the publication of findings in peer-reviewed journals and conferences, the development of open-source tools and datasets to support further research in the field, and collaborations with industry partners to promote responsible and ethical AI use in content moderation.

The successful implementation and deployment of the proposed system have the potential to transform cyberbullying detection and mitigation strategies, offering a scalable and effective solution to a pervasive social problem.

5. SYSTEM REQUIREMENTS

5.1 Hardware Requirements:

- System Type : intel®core™i3-7500UCPU@2.40gh
- Cache memory : 4MB(Megabyte)
- RAM : 8GB (gigabyte)
- Hard Disk : 256GB or more

5.2 Software Requirements:

- Operating System : Windows 11(Version 24H2)
- Coding Language : Python
- Python distribution : Anaconda, Flask, Google Colab
- Browser : Any Latest Browser like Chrome

6. SYSTEM ANALYSIS

6.1 Scope of the Project

This project aims to develop a deep learning-based multimodal system for cyberbullying detection across text and image data. Cyberbullying, which involves harassment via digital platforms, affects mental health and social well-being. Current detection methods predominantly focus on text-based content, leaving a gap in handling image-text combinations, often seen in memes. This research integrates pre-trained VGG16 for image analysis and XLM-RoBERTa with BiGRU for text processing, forming a multimodal framework. It leverages advanced techniques like CLIP, intermodal attention mechanisms, and CentralNet to capture contextual cues.

The primary objective is to improve detection accuracy by analyzing multimodal data comprehensively. The project emphasizes preprocessing, feature extraction, and classification to build a scalable, robust model that helps create safer online environments by effectively identifying bullying patterns across social media posts.

6.2 Dataset Description

The dataset used in this project is a publicly available multimodal social media dataset comprising text, images, and memes, specifically curated for cyberbullying detection. It contains a total of 6,006 samples, which, after preprocessing, were reduced to 5,798 posts, including 3,193 labeled as bully and 2,605 as non-bully as shown in Table 6.1 . Each data sample consists of an image and its corresponding text caption or comment, allowing the model to analyze both visual and textual cues for detecting cyberbullying.

The text data includes captions, comments, or meme text collected from social media platforms, while the image data consists of memes or other visual content often shared online. The dataset is labeled based on whether the post conveys bullying or abusive content, annotated through expert labeling or crowdsourced methods.

Preprocessing steps were applied to both modalities. Text data underwent tokenization, lowercasing, and removal of special characters, while images were

resized (224x224), normalized, and augmented (rotation, flipping) to improve model generalization. The dataset was split into 80% for training and 20% for testing.

This multimodal dataset reflects real-world scenarios where cyberbullying is communicated through both images and text, providing a comprehensive resource for training deep learning models aimed at cyberbullying detection.

S.No	Category	Total Samples	Bully Samples	Non-Bully Samples	Train Samples (80%)	Test Samples (20%)
1	Before Preprocessing	6,006	3,300	2,706	4,804	1,202
2	After Preprocessing	5,798	3,193	2,605	4,638	1,160

Table 6.1: Distribution of Images

6.3 Analysis

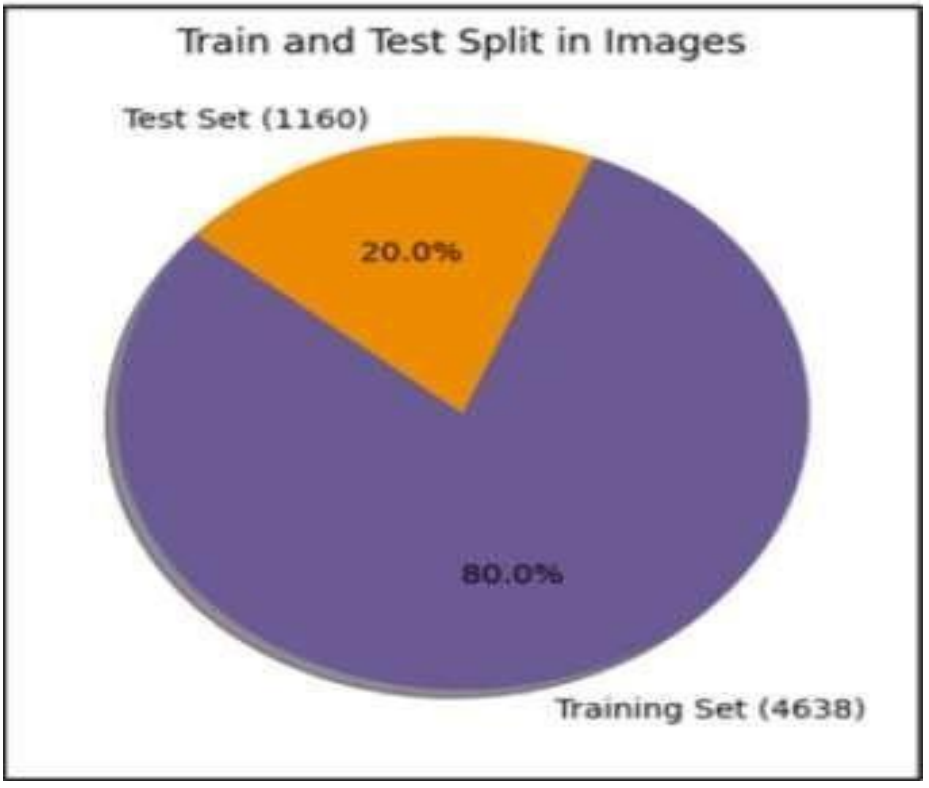

Fig 6.1 Train and Test Split In Images

The analysis phase Figure 6.1 explores the challenges of detecting cyberbullying, considering its complex multimodal nature. Previous research highlights effective machine learning approaches for text and image-based bullying detection. Text models, such as XLM-RoBERTa and BiGRU, demonstrate strong linguistic analysis, while image classifiers like VGG16 excel in visual pattern recognition. However, these methods lack synergy in handling multimodal data. The proposed approach merges these strengths, enabling detection of implicit bullying conveyed through memes. Dataset analysis Figure 6.2 reveals balanced distribution between bully and non-bully classes, ensuring model generalization. Metrics such as precision, recall, and F1-score validate model performance, addressing multimodal complexities with improved interpretability.

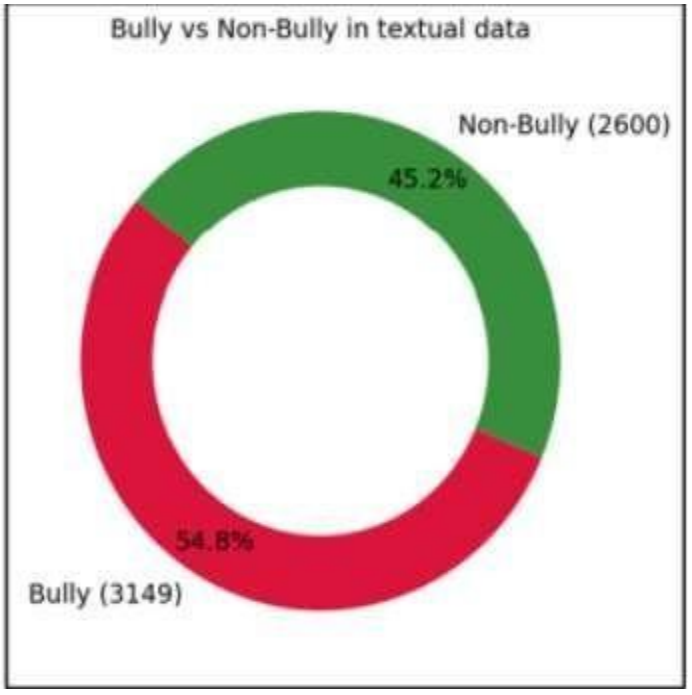

Fig 6.2 Bully Vs Non-Bully in Textual data

6.4 Data Pre-processing

Data pre-processing ensures consistency and quality for both image and text inputs. Image data undergoes normalization at Figure 6.3 using the Python Imaging Library (PIL) and Figure 6.4 augmentation via ImageData Generator to expand dataset diversity. Techniques like rotation, and flipping enhance robustness. Images are resized (224x224) for compatibility with pre-trained VGG16 models.

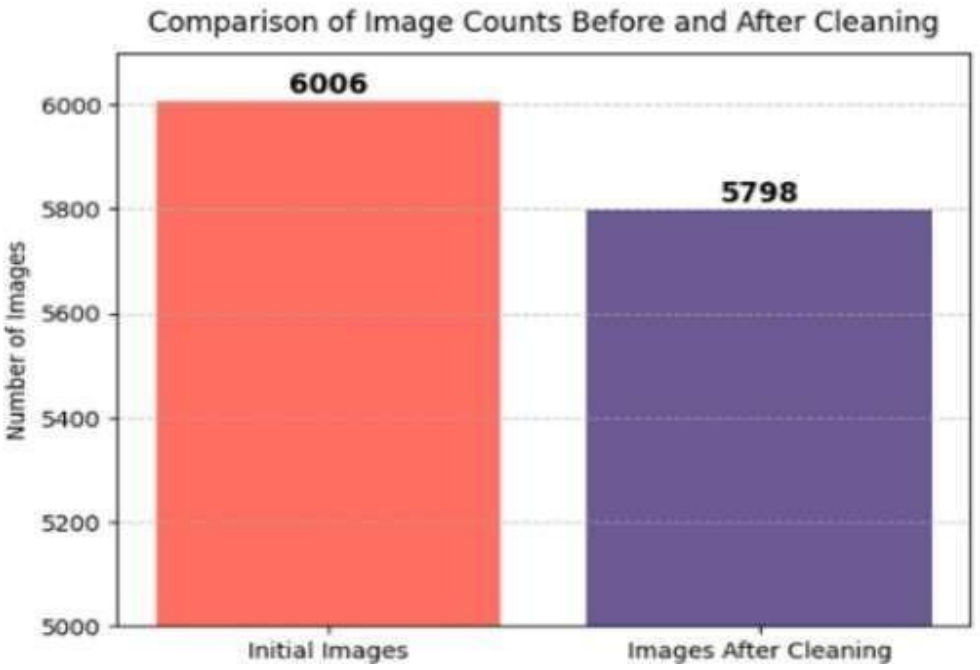

Fig 6.3 Comparison of Image Counts Before and After Cleansing

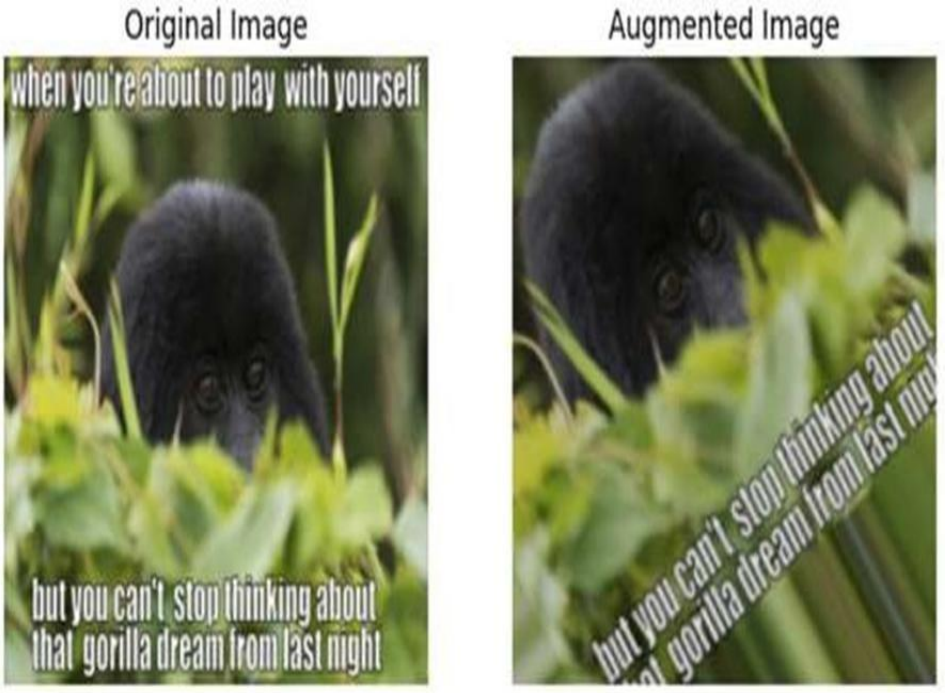

Fig 6.4 Image Augmentation

Text data preprocessing involves loading, cleaning, and structuring data using pandas. Unnecessary columns, duplicates, and null values are removed, reducing entries from 5865 to 5749(Figure 6.5).

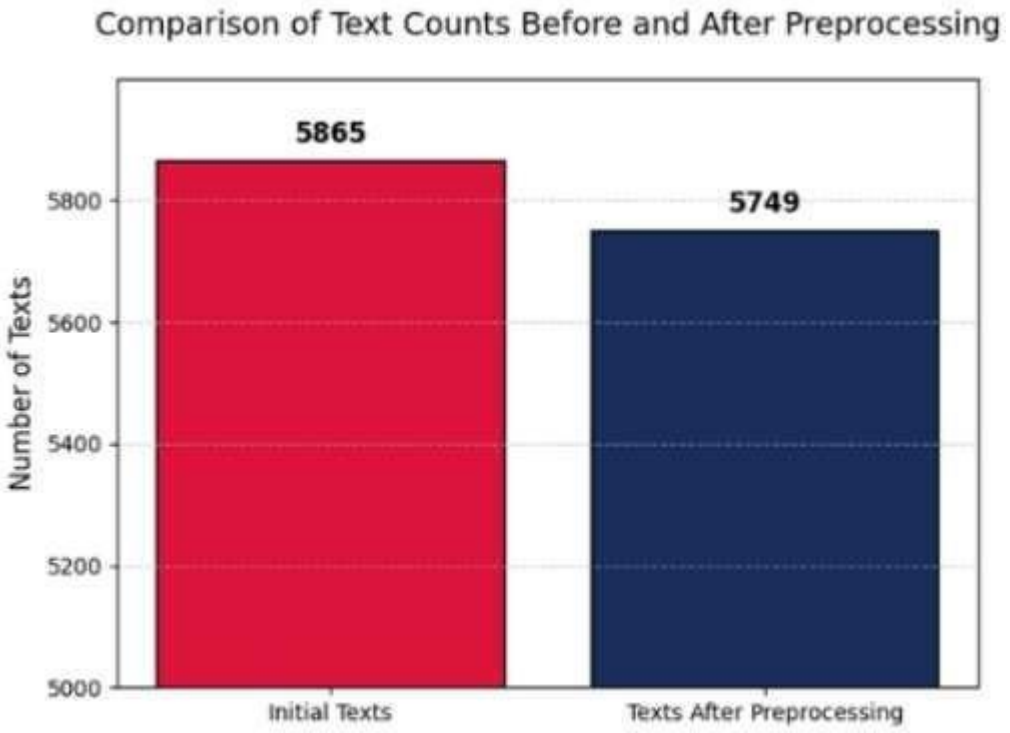

Fig 6.5 Comparison of Text Counts Before and After Preprocessing

Label encoding converts categories into numeric values, and tokenization using XLM- RoBERTaTokenizer splits text into manageable subword units. Further, labels are converted to tensors for compatibility with deep learning frameworks. Train-test splits (80:20) for both modalities ensure unbiased evaluation as shown in Figure 6.6 . Final datasets consist of 5798 labeled posts, equally distributed across bully and non-bully categories. Proper pre-processing avoids overfitting, boosts model performance, and enables the integration of text and image inputs.

	Img_Text	Img_Text_Label
0	Shivam @shivamishraa Girls be named naina and ...	0
1	Aaloo ke paranthe is the best breakfast Omelet...	1
2	For Boyfriend For Bestfriend DESI ADUKT TROLLS	0
3	You find a new YouTuber He's funny All of his ...	1
4	not_shubham14 @mentally_dank Kids at Marine Dr...	0

Fig 6.6 After Label Encoding

6.5 Feature Extraction:

Feature extraction focuses on leveraging pre-trained architectures for multimodal inputs. For images, VGG16 captures high-level visual features, including edges, shapes, and textures. The final layers are fine-tuned using dropout and dense layers to adapt to cyberbullying-specific patterns. Text data employs XLM-RoBERTa, which generates contextual embeddings from linguistic patterns, while BiGRU adds sequential learning capabilities for long-term dependencies. Intermodal attention mechanisms fuse these features by aligning visual and textual contexts, enabling deeper semantic understanding. CLIP projectors further refine cross-modal alignment by mapping text and image features into a shared space. The integration of fully connected layers consolidates features into a unified representation, facilitating bullying classification. These techniques ensure comprehensive analysis of multimodal data, identifying implicit and explicit bullying cues with improved generalization across diverse datasets.

6.6 Model Building:

The proposed model integrates advanced deep learning techniques to detect cyberbullying in multimodal data, leveraging both text and image inputs. It combines VGG16 for image analysis and XLM-RoBERTa with BiGRU for text processing, forming a robust multimodal framework.

6.6.1 Image Model

For image-based detection, VGG16, a pre-trained convolutional neural network (CNN), is employed. It extracts spatial features such as edges, textures, and shapes. The model's fully connected layers are customized, incorporating dropout (rate = 0.5) to prevent overfitting. The Global Average Pooling layer reduces dimensionality, and dense layers are added for classification. The Adam optimizer, with a learning rate of 1e-5, ensures efficient training, while cross-entropy loss is used for error evaluation.

Traditional methods that rely solely on textual analysis often fail to detect harmful messages embedded in images, memes, and other visual content. This project employs deep learning techniques, specifically leveraging the VGG16 model, to classify images as bully or non-bully based on visual cues. The flowchart

Figure 6.7 illustrates the step-by-step approach taken in this research, from data preprocessing to model fine-tuning and classification.

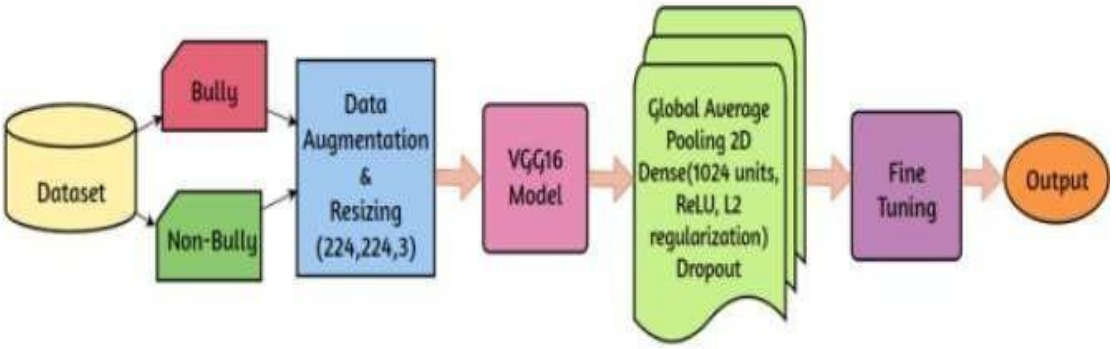

Fig 6.7 Image Based Detection

1. Dataset

The first stage in the pipeline involves collecting a dataset that contains labeled images categorized into two classes:

- Bully: Images that contain offensive, harmful, or bullying content.
- Non-Bully: Neutral or safe images that do not contain cyberbullying elements.

The dataset is a crucial component of the system, as it provides the necessary input for training the deep learning model. In this study, datasets are sourced from publicly available cyberbullying datasets, social media platforms, and curated image repositories containing labeled meme data. Data labeling is performed either manually or using pre-annotated sources.

2. Data Augmentation & Resizing

Since deep learning models require a significant amount of data to generalize well, data augmentation techniques are applied to artificially expand the dataset. This helps prevent overfitting and ensures the model learns meaningful patterns rather than memorizing specific images.

The augmentation techniques used include:

- Rotation: Randomly rotating images to make the model invariant to orientation changes.
- Flipping: Applying horizontal flips to introduce variation in image representation.

- Zooming: Random zooming to simulate different perspectives.
- Brightness Adjustments: Modifying brightness levels to ensure the model adapts to different lighting conditions.
- Contrast Variations: Increasing or decreasing contrast to account for diverse image qualities.

After augmentation, all images are resized to a fixed dimension of (224,224,3), which matches the input requirement of the VGG16 model. The three channels correspond to the RGB color space.

3. Feature Extraction Using VGG16 Model

The VGG16 model, a deep convolutional neural network (CNN), is used as the feature extractor in this study. It has been pre-trained on ImageNet, making it highly effective at recognizing visual patterns. The architecture consists of 16 layers, including convolutional layers, pooling layers, and fully connected layers.

When an image is passed through VGG16, the early convolutional layers capture low-level features such as edges, textures, and patterns. As the data progresses deeper into the network, it captures more high-level semantic features, which are crucial for identifying bullying-related content.

The advantages of using VGG16 include:

- Pre-trained Weights: Reduces training time and enhances performance.
- Hierarchical Feature Learning: Captures complex patterns that distinguish bullying images.
- Transfer Learning Capability: Can be fine-tuned for specific tasks, making it adaptable to cyberbullying detection

4. Global Average Pooling & Dense Layers

After feature extraction, the Global Average Pooling (GAP) layer is applied to reduce the spatial dimensions of feature maps, converting them into a one-dimensional feature vector. GAP is used instead of fully connected layers to reduce model complexity, thereby decreasing the chances of overfitting.

A dense (fully connected) layer follows, consisting of 1024 neurons, using the

ReLU (Rectified Linear Unit) activation function and L2 regularization to prevent overfitting. L2 regularization ensures that the model does not become overly dependent on specific features, improving generalization on unseen data.

Dropout regularization is also applied, randomly setting a fraction of neurons to zero during training to improve model robustness and prevent co-adaptation of features.

5. Fine-Tuning

Fine-tuning is a critical step where pre-trained layers of VGG16 are selectively unfrozen and retrained on the cyberbullying dataset. This process allows the model to adapt to the specific characteristics of bullying-related images while retaining the generalization capability of its original training on ImageNet.

During fine-tuning:

- Lower convolutional layers remain frozen, preserving fundamental edge and texture detection capabilities.
- Higher layers are unfrozen, allowing the model to learn dataset-specific patterns.
- A lower learning rate is used to prevent drastic weight updates that could erase learned features.

Fine-tuning enhances model performance and ensures it is optimized for cyberbullying detection without requiring an extremely large dataset.

6. Output Layer

The final layer in the architecture is the output layer, which classifies the image as bully or non-bully based on extracted features. It consists of:

- Two neurons, representing the two classes (bully and non-bully).
- Softmax activation function, which converts output scores into probability distributions.
- Cross-entropy loss function, used for multi-class classification tasks to measure prediction accuracy.

The model outputs a probability score for each class, with the highest probability determining the final classification.

7. Model Performance Evaluation

To assess the effectiveness of the cyberbullying detection system, several evaluation metrics are used:

- Accuracy: Measures the percentage of correctly classified images.
- Precision: Evaluates the proportion of true positive predictions among all positive classifications.
- Recall: Assesses the model's ability to detect all actual bullying images.
- F1-Score: A balanced measure combining precision and recall.
- Confusion Matrix: Visualizes model performance by showing true positives, false positives, true negatives, and false negatives.

This Deep Learning-based approach is valuable for detecting cyberbullying in images using the **VGG16 model**. By leveraging data augmentation, feature extraction, global average pooling, dropout regularization, and fine-tuning, the system achieves robust classification performance.

6.6.2 Text Model

For text-based detection, XLM-RoBERTa, a transformer-based language model, generates contextual embeddings. It handles multilingual data and complex sentence structures effectively. The embeddings are passed through a BiGRU (Bidirectional Gated Recurrent Unit), which captures sequential dependencies. Dropout and batch normalization prevent overfitting, while AdamW optimizer fine-tunes weights dynamically. We employed deep learning-based natural language processing (NLP) techniques to detect cyberbullying in textual data effectively.

1. Input Text

The process starts with raw textual input from various online sources such as:

- **Social media posts** (tweets, comments, captions, etc.)
- **Chat messages**
- **Online forum discussions**
- **Email or private messages**

This text serves as the primary input for the cyberbullying detection system. Since

language varies significantly across different platforms and users, preprocessing techniques are applied to ensure consistency and quality in the input data as shown in Figure 6.8.

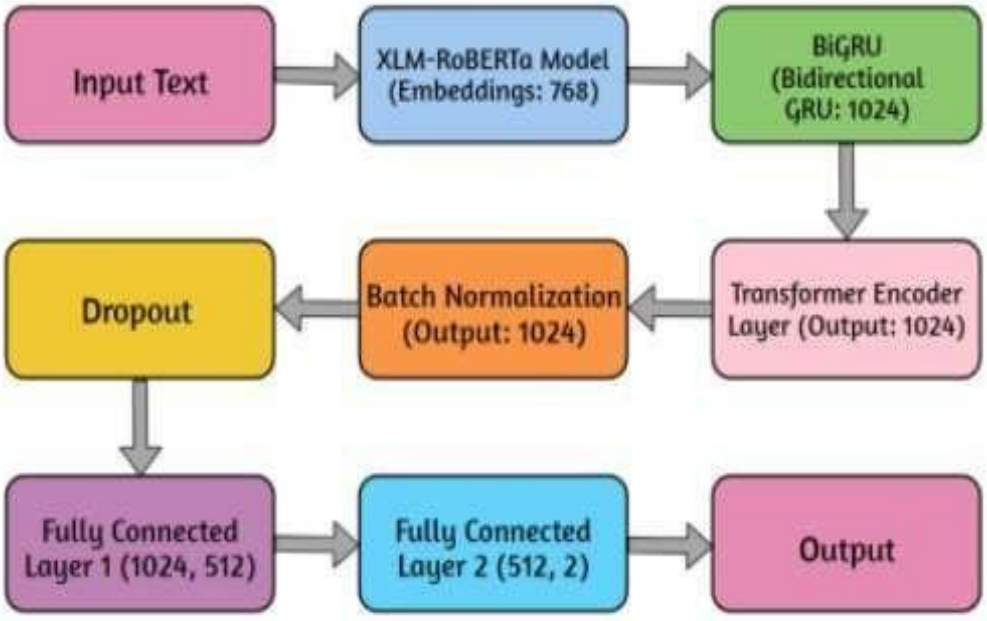

Fig 6.8 Text Based Detection

Preprocessing steps include:

- **Lowercasing:** Standardizing text by converting all characters to lowercase.
- **Removing special characters and punctuation:** Eliminating symbols that do not contribute to semantic meaning.
- **Stopword removal:** Filtering out common words (e.g., "the," "and") that do not provide significant contextual information.
- **Tokenization:** Breaking text into individual words or subwords for processing.
- **Normalization:** Handling variations in text by converting slang, abbreviations, or misspellings to their standard forms.

Once preprocessed, the text is fed into the **XLM-RoBERTa model**, which converts it into dense numerical representations known as embeddings.

2. XLM-RoBERTa Model (Embeddings: 768)

XLM-RoBERTa (Cross-lingual Language Model - RoBERTa) is a transformer-based NLP model trained on multiple languages. It extends RoBERTa's architecture and is designed to process diverse linguistic patterns, making it highly suitable for detecting cyberbullying in multilingual environments.

The XLM-RoBERTa model takes preprocessed text as input and generates word embeddings—vector representations of words—where similar words have similar numerical representations. Each input text is transformed into an embedding of size 768 dimensions, representing semantic and syntactic meanings.

Key Advantages of Using XLM-RoBERTa:

- **Multilingual Capability:** Effective for handling multiple languages and code-mixed text.
- **Contextual Understanding:** Unlike traditional word embeddings (Word2Vec, GloVe), XLM-RoBERTa captures contextual meaning, understanding words differently based on surrounding text.
- **Transfer Learning:** The model is pre-trained on a vast amount of text data and can be fine-tuned for specific tasks like cyberbullying detection.

The **768-dimensional embeddings** from XLM-RoBERTa are passed into a **Bidirectional GRU (BiGRU)** for further processing.

3. BiGRU (Bidirectional GRU: 1024)

The **Bidirectional Gated Recurrent Unit (BiGRU)** is a variant of the **Gated Recurrent Unit** (GRU), which is a type of recurrent neural network (RNN). BiGRU processes input text in both forward and backward directions, capturing dependencies between words effectively.

Why Use BiGRU?

- **Bidirectionality:** Unlike standard GRUs, which process sequences only from left to right, BiGRUs capture past and future context by processing sequences in both directions.
- **Efficient Memory Usage:** Compared to Long Short-Term Memory (LSTM) networks, GRUs have fewer parameters, making them more computationally

efficient while maintaining similar performance.

- **Better Context Retention:** Essential for detecting cyberbullying where meaning often depends on surrounding words (e.g., sarcasm, double meanings).

The 1024-dimensional output from BiGRU is then passed to a Transformer Encoder Layer for further refinement.

4. Transformer Encoder Layer (Output: 1024)

The Transformer Encoder Layer further enhances feature extraction by applying self-attention mechanisms to focus on relevant parts of the text. This layer refines the 1024-dimensional features generated by BiGRU.

How the Transformer Encoder Works:

- **Self-Attention Mechanism:** Assigns weights to different words based on their importance in the sentence.
- **Positional Encoding:** Retains word order information, which is critical for understanding sentence structure.
- **Layer Normalization:** Ensures stable learning and prevents exploding gradients.

This layer helps in detecting hidden patterns in bullying-related text, such as sarcasm, offensive language, and indirect bullying phrases. The refined features are then normalized before classification.

5. Batch Normalization (Output: 1024)

Batch normalization stabilizes the learning process by normalizing the transformed features. It helps:

- Improve convergence speed.
- Reduce internal covariate shifts.
- Prevent overfitting.

The normalized 1024-dimensional features are then subjected to Dropout regularization to further enhance model generalization.

6. Dropout Regularization

Dropout is a regularization technique that randomly drops a fraction of neurons during training. This prevents the model from overfitting to training data and ensures better generalization to unseen text.

In this architecture, dropout is applied before the fully connected layers, ensuring that the model does not become overly dependent on specific neurons.

7. Fully Connected Layers (1024 → 512 → 2)

The final classification process involves two fully connected (dense) layers:

1. **Fully Connected Layer 1 (1024, 512):**
    - Applies ReLU activation for non-linearity.
    - Reduces feature dimensions from 1024 to 512.
2. **Fully Connected Layer 2 (512, 2):**
    - Maps 512 features to 2 output neurons.
    - Uses a Softmax activation function to predict probabilities for the two classes (bully or non-bully).

8. Output Layer (Final Classification)

The final output consists of:

- Two neurons, representing the two classes:
    - **Bully (1)**
    - **Non-Bully (0)**
- Softmax activation function, which converts raw scores into probability values.
- The class with the highest probability is chosen as the final prediction.

6.6.3 Multimodal Fusion

To combine textual and visual features, the model utilizes CLIP projectors and intermodal attention mechanisms. CLIP projectors map text and image features into a shared embedding space, enabling effective alignment of multimodal data. Intermodal attention layers emphasize contextual relationships between visual and textual features, improving semantic understanding.

The CentralNet framework merges outputs from image-text models, aligning them into unified feature representation. Fully connected layers refine the combined embeddings, and a softmax classifier outputs binary labels bully or non-bully.

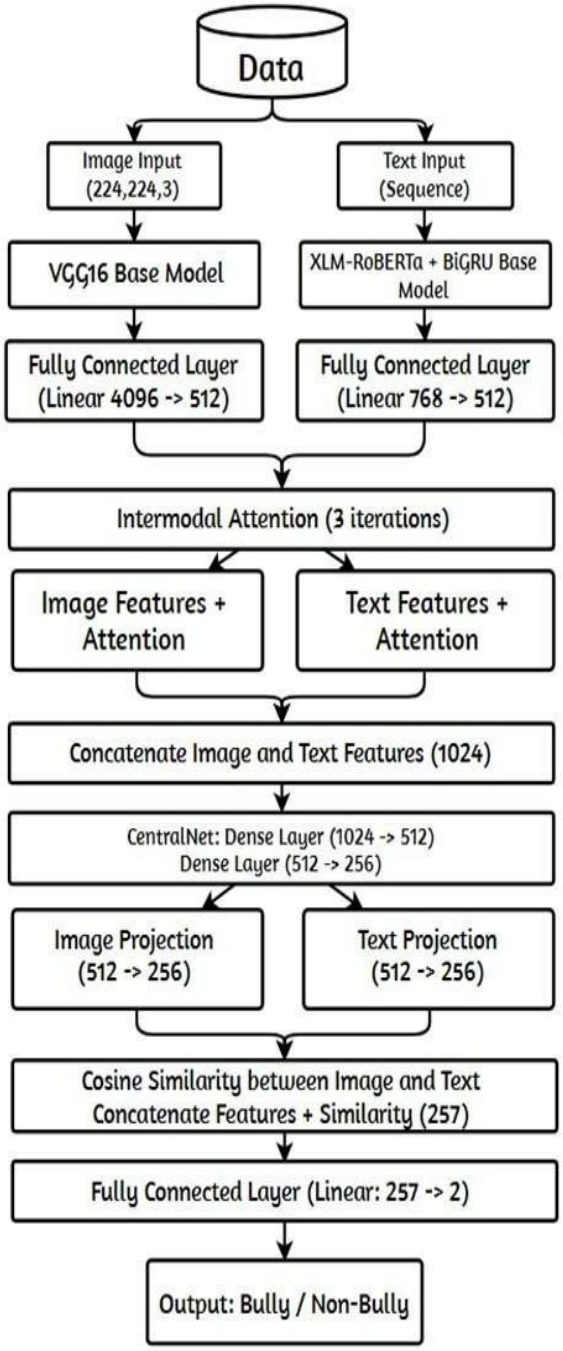

Fig 6.9 Multimodal Data Detection

The architecture comprises multiple stages as shown in Figure 6.9 :

1. Data Inputs: Image and Text

The input data for this model consists of two primary components:

1. **Image Input:** The images, often memes or other visual content, are resized to **(224, 224, 3)** to match the input size of **VGG16**, a well-known Convolutional

Neural Network (CNN) used for feature extraction.

2. **Text Input:** The textual data, which could be captions, comments, or embedded text within memes, is processed as a **sequence** of tokens. This is handled using the **XLM-RoBERTa** transformer-based model combined with a **Bidirectional Gated Recurrent Unit (BiGRU)** to capture the contextual meaning of words.

2. Feature Extraction using Pretrained Models

After receiving the inputs, the model extracts meaningful features from both modalities using **deep learning architectures**:

- **VGG16 Base Model (for images):**
    - A CNN-based architecture that captures deep hierarchical patterns in images.
    - The **fully connected layer** transforms high-dimensional features from **4096 to 512** dimensions for better compatibility with text features.
- **XLM-RoBERTa + BiGRU (for text):**
    - **XLM-RoBERTa** is a multilingual variant of the **RoBERTa** model, trained for diverse linguistic contexts.
    - **BiGRU (Bidirectional GRU)** is used to retain long-term dependencies in the text.
    - A **fully connected layer** reduces the text feature dimensions from **768 to 512** to align with image features.

3. Intermodal Attention Mechanism

The extracted features from **both images and text** are then **processed using an intermodal attention mechanism**. This attention mechanism works for **3 iterations**, ensuring that the model effectively focuses on the most important regions of the image and the most significant parts of the text.

- **Image Features + Attention:** Enhances the image representation by giving more weight to the **most relevant regions**.
- **Text Features + Attention:** Helps the model **attend to specific words** or phrases crucial for cyberbullying detection.

4. Fusion of Multimodal Features

Once the attention-enhanced features are extracted, they are **concatenated into a unified multimodal representation** of size **1024**. This fusion allows the model to understand the **interplay between text and images**, which is crucial for detecting complex bullying cases where meaning is conveyed through both modalities.

- **CentralNet Dense Layers:**
    - A dense layer reduces the **1024-dimensional feature vector** to **512**.
    - Another dense layer further reduces it to **256**, making the representation more compact while retaining meaningful relationships.

5. Projection of Image and Text Features

To further refine the multimodal representation:

- The **image feature vector** is projected into a **256-dimensional space**.
- Similarly, the **text feature vector** is projected into the **same 256-dimensional space**.
- This projection helps in ensuring that both modalities are represented in a comparable manner, facilitating **cross-modal understanding**.

6. Cosine Similarity and Feature Concatenation

After projecting the image and text features into the same space, the model computes the **cosine similarity** between them. **Cosine similarity** is used to measure how closely related the image and text representations are. This is crucial in multimodal cyberbullying detection because:

- A high similarity score could indicate **harmonious content**, meaning the text and image are related in a neutral or positive way.
- A low similarity score could indicate **misalignment**, often a sign of cyberbullying, where text and images are used in an offensive manner.

The final multimodal feature representation is constructed by **concatenating the cosine similarity score with the previous feature vector**, leading to a **257-dimensional representation** (256 from the concatenated features + 1 from the similarity score).

7. Fully Connected Classification Layer

The final step involves a **fully connected layer** that processes the **257-dimensional** feature vector and maps it to **two output classes: Bully or Non-Bully**. This classification layer is responsible for making the final prediction by analyzing the extracted multimodal features.

8. Output: Cyberbullying Classification

The final model output is a **binary classification**:

- **Bully:** If the post contains harmful, offensive, or aggressive content.
- **Non-Bully:** If the post is neutral or non-offensive.

This prediction is based on the **intermodal attention, cosine similarity, and dense layers**, ensuring a well-informed decision-making process.

6.7 Classification

Classification is the final and most critical stage in the proposed multimodal cyberbullying detection model. It involves integrating features extracted from text and image data, fusing them into a unified representation, and passing them through dense layers for prediction. This section details the classification process, covering the techniques, layers, and mechanisms that enhance the model's performance.

6.7.1 Multimodal Input Handling

The classification process begins with multimodal inputs comprising both text and images. These inputs are preprocessed to ensure compatibility with deep learning models:

Text Preprocessing: Tokenization, label encoding, and tensor conversion prepare textual data for embedding models.

Image Preprocessing: Resizing, normalization, and augmentation improve visual data quality and consistency.

The preprocessed inputs are then fed into separate models for feature extraction: VGG16 processes image inputs to extract spatial features.

XLM-RoBERTa with BiGRU analyzes textual data, capturing contextual and sequential patterns.

6.7.2 Feature Extraction and Embedding Generation

Image Feature Extraction:

The VGG16 pre-trained model processes images through convolutional layers, identifying visual patterns such as shapes, edges, and textures.

A Global Average Pooling layer reduces dimensionality without losing essential information. Dropout layers (rate = 0.5) prevent overfitting during training. Dense layers map features into a lower-dimensional vector space for compatibility with textual features.

Text Feature Extraction:

Textual features are extracted using XLM-RoBERTa, a transformer-based model that generates contextual embeddings.

BiGRU layers are added to handle sequential dependencies, enhancing pattern recognition in text sequences.

Dropout layers and batch normalization ensure regularization and stability during training.

Both sets of features are projected into a unified embedding space using CLIP

projectors, ensuring compatibility between modalities.

6.7.3 Feature Fusion

The multimodal integration process combines features from text and image modalities, allowing the model to analyze their relationships.

CLIP Projectors:

Map features into a shared space to facilitate alignment between visual and textual features. Enable better semantic understanding by associating image elements with related text components.

Intermodal Attention Mechanism:

Employs multiple attention heads to focus on relationships between specific visual and textual features.

Enhances model performance by highlighting relevant patterns and suppressing noise.

CentralNet Framework:

Aligns multimodal embeddings through hierarchical connections. Refines features by emphasizing shared attributes across modalities.

6.7.4 Fully Connected Layers

Once the features are fused, they pass through fully connected (dense) layers to produce the final prediction.

Layer Configuration:

- Input Size: 512 neurons (combined features).
- Hidden Layers: Multiple dense layers with ReLU activation functions.
- Dropout Layers: Applied after each hidden layer to reduce overfitting.
- Output Layer: A softmax classifier outputs probabilities for two classes bully or non bully

Loss Function:

Cross-Entropy Loss calculates errors during training and adjusts weights to minimize classification errors.

Optimizer:

AdamW Optimizer dynamically adjusts learning rates, ensuring stable and efficient convergence.

6.7.5 Training and Hyperparameter Tuning

The classification model is trained using an 80:20 split of the dataset.

Key Hyperparameters:

- Batch Size: 32 (for balanced memory usage stability).
- Learning Rate: 1e-5 (fine-tuned for faster convergence).
- Weight Decay: 0.01 (for regularization).
- Early Stopping: Stops training after 3 consecutive epochs without performance improvement. During training, weights are updated iteratively using gradient descent, optimizing performance based on loss values.

6.7.6 Evaluation Metrics

To assess classification performance, the Accuracy, Precision, Recall, and F1 Score metrics are calculated.

Confusion Matrix: Visualizes prediction distribution, including true positives, true negatives, false positives, and false negatives.

6.7.7 Results and Analysis

Image Model Performance (VGG16):

- Accuracy: 64.74%
- Precision: Moderate, effective in identifying image patterns but limited by visual ambiguities.

Text Model Performance (XLM-RoBERTa + BiGRU):

- Accuracy: 64.78%
- High recall, capturing bullying patterns in textual content but requiring visual context for implicit cues.

Multimodal Model Performance (VGG16 + XLM-RoBERTa + BiGRU):

- Accuracy: 74%
- Significantly outperforms single-modal models by leveraging complementary visual and textual information.
- Achieves better precision and recall due to feature fusion and attention mechanisms.

6.7.8 Error Analysis

Despite its high performance, some errors persist, primarily in:

Implicit Bullying Patterns: Sarcasm and humor can be challenging to classify.

Ambiguous Images: Contextual ambiguity in visual elements may cause misclassification. Language Variations: Code-mixed and multilingual content may introduce noise.

Solutions for Future Models:

- Incorporate advanced language transformers like GPT or multilingual embeddings.
- Use GAN-based image augmentation for richer training data.
- Introduce adversarial training to handle ambiguity better.

6.7.9 Scalability and Deployment

The model is scalable and can be deployed on cloud platforms for real-time analysis.

Key Features for Deployment:

Edge Computing Compatibility: Lightweight architectures for mobile and

web integration. API Integration: Seamless deployment in social media moderation systems.

Alert Mechanisms: Automated notifications for flagged content.

6.8 Confusion Matrix

A Confusion Matrix is a valuable and effective tool used for evaluating classification models. Confusion matrix provides a detailed summary of the model's predictions. This summary enables the assessment of key performance metrics. Accuracy, precision, recall, and F1-score are commonly evaluated. The Confusion Matrix offers insights into true positives, false positives, true negatives, and false negatives. By analyzing these metrics, model performance can be optimized. This leads to improved decision-making and more accurate predictions. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which true values are known.

True Positives (TP) and True Negatives (TN)

True positives occur when your model correctly predicts a positive outcome whereas True negatives occur when your model correctly predicts a negative outcome.

False Postives (FP) and False Negatives (FN)

False positives occur when your model incorrectly predicts a positive outcome whereas False negatives occur when your model incorrectly predicts a negative ooutcome.

Performance Metrics Derived from Confusion Matrix :

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

F1-score = 2 * (Precision * Recall) / (Precision + Recall)

The results of Multimodal indicate that our proposed model achieves a decent accuracy of 74% in detecting cyberbullying in multimodal data.

7. DESIGN

The proposed model is a multimodal cyberbullying detection system that integrates both image and text data to identify instances of cyberbullying as mentioned in Table 7.1 . The model consists of two primary components. An image-based model and a text-based model. The image-based model utilizes a pre-trained VGG16 architecture, which has been fine-tuned for cyberbullying detection. The model takes images as input, extracts relevant features using convolutional neural network (CNN) layers, and then uses fully connected layers to classify the images as either bully or non-bully.

Component	**Hyperparameter**	**Value**
Image Model (VGG16)	Dropout Rate Fully Connected Layer	0.5 512 (from 25088)
Text Model (XLM-RoBERTa)	Fully Connected Layer Output Size	512 (from 768)
CLIP projectors	Image Projector Text Projector	256 (from 512) 256 (from 512)
Intermodal Attention	Embedding Dimension Number of Attention Heads	512 8
Final Classifier	Output Size Number of Classes	257 (256 + 1) 2
Training Settings	Batch Size Learning Rate Weight Decay Early Stopping Patience Loss Function Optimizer	32 1e-5 0.01 3 Cross Entropy AdamW

Table 7.1: Hyperparameters for Multimodal

The text-based model employs a pre-trained XLM-RoBERTa architecture, which has been fine-tuned for cyberbullying detection. The model takes text data as input, tokenizes the text using a special tokenizer, and then uses a bidirectional GRU (BiGRU) layer to extract contextual features. The output from the BiGRU layer is then fed into a fully connected layer to classify the text as either bully or non-bully. This approach enables the model to capture complex patterns and relationships in the

text data.

The image-based and text-based models are combined using a multimodal fusion approach. The features extracted from the image-based model are concatenated with the features extracted from the text-based model. The concatenated features are then fed into a fully connected layer to produce the final output. This multimodal fusion approach enables the model to leverage the strengths of both image and text data, resulting in improved performance and accuracy.

The model is trained using a dataset consisting of images and text data. The dataset is split into training and testing sets, with 80% of the data used for training and 20% used for testing. The model is trained using the AdamW optimizer and a learning rate of 1e-5. The model is also regularized using dropout and weight decay. This approach enables the model to learn effective representations of the data and generalize well to new, unseen instances.

8. IMPLEMENTATION

Image Modal Code

```
from google.colab import drive
drive.mount('/content/drive')

import tensorflow as tf
from tensorflow.keras.preprocessing.image import
ImageDataGenerator from tensorflow.keras.applications
import VGG16
from tensorflow.keras.layers import Dense,
GlobalAveragePooling2D, Dropout from
tensorflow.keras.models import Model
from tensorflow.keras.callbacks import ReduceLROnPlateau, EarlyStopping,
ModelCheckpoint

# Paths to your dataset train_dir =
'/content/drive/MyDrive/train' # Update with your path to
training data test_dir = '/content/drive/MyDrive/test' # Update
with your path to test data

# ImageDataGenerator for data
augmentation and normalization
train_datagen = ImageDataGenerator(
rescale=1./255, rotation_range=40,
width_shift_range=0.2,
height_shift_range=0.2, shear_range=0.2,
zoom_range=0.2, horizontal_flip=True,
fill_mode='nearest',
    brightness_range=[0.8, 1.2]
)

test_datagen = ImageDataGenerator(rescale=1./255)

# Flow training images in batches of 32 using
train_datagen generator train_generator =
train_datagen.flow_from_directory( train_dir,
target_size=(224, 224), # Resize images to (224,
224) for VGG16 batch_size=32,
    class_mode='binary' # For binary classification
)
```

```
# Flow validation images in batches of 32 using
test_datagen generator test_generator =
test_datagen.flow_from_directory( test_dir,
    target_size=(224,
    224),
    batch_size=32,
    class_mode='binary'
)

# Load VGG16 with pre-trained ImageNet weights, excluding top layers
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(224,
224, 3))

# Add custom top layers for binary classification
x =
base_model.o
utput x =
GlobalAverag
ePooling2D()
(x)
x = Dense(1024, activation='relu',
kernel_regularizer=tf.keras.regularizers.l2(0.001))(x) x =
Dropout(0.5)(x)
predictions = Dense(1, activation='sigmoid')(x)

# Combine base model and top layers into
new model model =
Model(inputs=base_model.input,
outputs=predictions)

# Freeze all layers of the base model
(only train top layers) for layer in
base_model.layers:
    layer.trainable = False

# Compile the model with Adam optimizer and a learning
rate of 0.001 optimizer =
tf.keras.optimizers.Adam(learning_rate=0.001)
model.compile(optimizer=optimizer, loss='binary_crossentropy',
metrics=['accuracy'])

# Callbacks for learning rate reduction, early stopping, and model
checkpoint reduce_lr = ReduceLROnPlateau(monitor='val_loss',
factor=0.2, patience=3, min_lr=1e-7) early_stopping =
```

```
EarlyStopping(monitor='val_loss', patience=10,
restore_best_weights=True) model_checkpoint =
ModelCheckpoint('/content/vgg16_bully_classifier_best.keras',
monitor='val_loss', save_best_only=True)
# Train the model on the data
history =
model.fit
(
train_ge
nerator,
    steps_per_epoch=train_generator.samples //
    train_generator.batch_size, epochs=40, # Initial
    training epochs validation_data=test_generator,
    validation_steps=test_generator.samples // test_generator.batch_size,
    callbacks=[reduce_lr, early_stopping, model_checkpoint]
)

# Unfreeze some of the top layers of the base
model for fine-tuning for layer in
base_model.layers[-100:]: # Unfreeze the last 50
layers layer.trainable = True

# Recompile the model with a lower learning rate for fine-
tuning optimizer_fine =
tf.keras.optimizers.Adam(learning_rate=1e-5)
model.compile(optimizer=optimizer_fine, loss='binary_crossentropy',
metrics=['accuracy'])

# Continue training the
model for fine-tuning
history_fine = model.fit(
train_generator,
    steps_per_epoch=train_generator.samples //
    train_generator.batch_size, epochs=40, #
    Additional fine-tuning epochs
    validation_data=test_generator,
    validation_steps=test_generator.samples // test_generator.batch_size,
    callbacks=[reduce_lr, early_stopping, model_checkpoint]
)

# Evaluate the model
test_loss, test_acc =
model.evaluate(test_generator) print(f'Test
accuracy: {test_acc}')
```

```
# Evaluate the model
test_loss, test_acc =
model.evaluate(test_generator) print(f'Test
accuracy: {test_acc}')
# Evaluate the model
train_loss, train_acc =
model.evaluate(train_generator) print(f'Train
accuracy: {train_acc}')

# Save the fine-tuned model
model.save('/content/drive/MyDrive/FINAL VGG16.keras')

import numpy as np from
tensorflow.keras.preprocessig
import image
from tensorflow.keras.models import load_model
import tensorflow as tf
model_path = '/content/drive/MyDrive/FINAL
VGG16.keras' model =
tf.keras.models.load_model(model_path)

# Evaluate the model
test_loss, test_acc =
model.evaluate(test_generator) print(f'Test
accuracy: {test_acc}')
```

Text Modal Code

```
# Install necessary libraries
# !pip install torch transformers scikit-learn tqdm nltk matplotlib seaborn

import pandas as pd
import numpy as np
import re
import torch nn as nn
from torch.utils.data import DataLoader, Dataset
from transformers import XLMRobertaTokenizer, XLMRobertaModel, AdamW,
get_linear_schedule_with_warmup
from sklearn.model_selection import
train_test_split from sklearn.preprocessing
import LabelEncoder from sklearn.metrics
import accuracy_score, confusion_matrix
from tqdm import tqdm import
```

```
matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
df = pd.read_csv('/content/cyberbullying dataset new.csv', encoding='latin-1')

# Drop unnecessary columns
 df.drop(['Unnamed: 5', 'Unnamed: 6', 'Unnamed: 7', 'Unnamed: 8', 'Unnamed: 9',
  'Unnamed: 10', 'Unnamed: 11', 'Img_Name', 'Img_Label', 'Text_Label'], axis=1,
                                                                    inplace=True)
# Drop duplicates and handle missing
values df.dropna(axis=0, inplace=True)
df.drop_duplicates(inplace=True)

# Encode labels
label_encoder = LabelEncoder()
df['Img_Text_Label'] = label_encoder.fit_transform(df['Img_Text_Label'])

# Initialize tokenizer for XLM-R
tokenizer_xlm = XLMRobertaTokenizer.from_pretrained('xlm-roberta-base')

# Tokenize texts for XLM-R
encoded_texts_xlm = tokenizer_xlm(df['Img_Text'].tolist(),
                                  padding=True,
                                  truncation=True,
                                  max_length=50,
                                  return_tensors='pt')
input_ids_xlm = encoded_texts_xlm['input_ids']
attention_mask_xlm = encoded_texts_xlm['attention_mask']

# Convert labels to tensor
labels = torch.tensor(df['Img_Text_Label'].values)

# Train-test split
train_inputs_xlm, val_inputs_xlm, train_labels, val_labels =
train_test_split(input_ids_xlm, labels, test_size=0.2, random_state=42)
train_masks_xlm, val_masks_xlm, _, _ = train_test_split(attention_mask_xlm,
labels, test_size=0.2, random_state=42)

# Define DataLoader class
CustomDataset(Dataset):
    def __init__(self, input_ids_xlm, attention_mask_xlm, labels):
                self.input_ids_xlm = input_ids_xlm
                self.attention_mask_xlm = attention_mask_xlm
```

```
                self.labels = labels
    def _len_(self): return
                len(self.input_ids_xlm)
     def getitem
        (self, idx):
        return (self.input_ids_xlm[idx], self.attention_mask_xlm[idx], self.labels[idx])

train_dataset = CustomDataset(train_inputs_xlm, train_masks_xlm,
train_labels) val_dataset = CustomDataset(val_inputs_xlm,
val_masks_xlm, val_labels)
train_loader = DataLoader(train_dataset, batch_size=16,
shuffle=True) val_loader = DataLoader(val_dataset,
batch_size=16, shuffle=False)

 # Define model architecture class
 CustomXLMGRUTransformer(nn.Module):
    def _init_(self, hidden_size=512, num_labels=2, dropout_rate=0.5):
        super(CustomXLMGRUTransformer, self)._init_()
        self.xlm_roberta = XLMRobertaModel.from_pretrained('xlm-roberta-base')

        # BiGRU layer
        self.gru = nn.GRU(input_size=768, hidden_size=hidden_size,
        batch_first=True,
   bidirectional=True)

        # Transformer layer
        self.transformer_layer = nn.TransformerEncoderLayer(d_model=hidden_size
        * 2, nhead=8)

        # Batch normalization
        self.batch_norm = nn.BatchNorm1d(hidden_size * 2)

        # Dropout layer
        self.dropout = nn.Dropout(dropout_rate)

        # Fully connected layers
        self.fc1 =
        nn.Linear(hidden_size * 2,
        hidden_size) self.fc2 =
        nn.Linear(hidden_size,
        num_labels)

    def forward(self, input_ids_xlm, attention_mask_xlm):
        # XLM-R embeddings
```

```
        xlm_outputs = self.xlm_roberta(input_ids_xlm,
        attention_mask=attention_mask_xlm) sequence_output =
        xlm_outputs.last_hidden_state # shape: (batch_size, 50, 768)

        # BiGRU
        gru_outputs, _ = self.gru(sequence_output) # shape: (batch_size, 50, 1024)

        # Transformer
        transformer_output = self.transformer_layer(gru_outputs) # shape: (batch_size,
        50, 1024)

        # Pooling
        pooled_output = torch.mean(transformer_output, dim=1) # shape: (batch_size,
        1024)

        # Batch normalization
        normalized_output = self.batch_norm(pooled_output)

        # Fully connected layers
        combined = self.dropout(normalized_output)
        combined =
        torch.relu(self.fc1(combined
        )) logits =
        self.fc2(combined)
        return logits

# Initialize the model
model = CustomXLMGRUTransformer()

# Print model architecture
# print(model)
# Define optimizer with weight decay
optimizer = AdamW(model.parameters(), lr=1e-5, eps=1e-8, weight_decay=0.01)

# Scheduler for learning rate
total_steps = len(train_loader) * 5 # 5 epochs
scheduler = get_linear_schedule_with_warmup(optimizer, num_warmup_steps=0,
num_training_steps=total_steps)

# Define loss function
criterion = nn.CrossEntropyLoss()

# Training loop
device = torch.device('cuda' if torch.cuda.is_available() else
```

```
'cpu') model.to(device)

epochs = 45
best_accuracy = 0

for epoch in range(epochs):
    # Training model.train() train_loss = 0 for batch
    in tqdm(train_loader, desc=f'Epoch {epoch +
    1}/{epochs}'): input_ids_xlm,
    attention_mask_xlm, labels = batch
        input_ids_xlm, attention_mask_xlm, labels = input_ids_xlm.to(device),
attention_mask_xlm.to(device), labels.to(device)

        optimizer.zero_grad()
        outputs = model(input_ids_xlm, attention_mask_xlm)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        scheduler.step()
        train_loss += loss.item()
    #validation
    model.eval()
    val_loss = 0
    val_predictions, val_true_labels = [], []
    with torch.no_grad():
        for batch in val_loader:
            input_ids_xlm,
            attention_mask_xlm, labels =
            batch
            input_ids_xlm, attention_mask_xlm, labels = input_ids_xlm.to(device),
attention_mask_xlm.to(device), labels.to(device)

            outputs = model(input_ids_xlm,
            attention_mask_xlm) loss =
            criterion(outputs, labels)

            val_loss += loss.item()
            preds =
            torch.argmax(outputs,
            dim=1)
            val_predictions.extend(
            preds.cpu().numpy())
            val_true_labels.extend(l
            abels.cpu().numpy())
```

```
    val_accuracy =
    accuracy_score(val_true_labels,
    val_predictions) if val_accuracy >
    best_accuracy: best_accuracy =
    val_accuracy
        torch.save(model, '/content/drive/MyDrive/final_xlm_bigru2.pth')

print(f"Epoch {epoch + 1}, Train Loss: {train_loss / len(train_loader):.4f}, Val
Loss: {val_loss / len(val_loader):.4f}, Val Accuracy: {val_accuracy:.4f}")

# Print the best accuracy after all epochs
print(f"Best Validation Accuracy: {best_accuracy:.4f}")
```

Flask Code to connect with Frontend

```
from flask import Flask, render_template,
request, jsonify import tensorflow as tf
from tensorflow.keras.preprocessing.image import load_img,
img_to_array import numpy as np
import os
from werkzeug.utils import secure_filename
import pytesseract from PIL import Image import
torch import torch.nn as nn from transformers
import XLMRobertaTokenizer,
XLMRobertaModel app = Flask(_name_)

# --------- Image Classification Setup ---------- #

Path to the saved image
classification model
IMAGE_MODEL_PATH = 'FINAL VGG16.keras'
image_model = tf.keras.models.load_model(IMAGE_MODEL_PATH)

# Path to Tesseract executable
pytesseract.pytesseract.tesseract_cmd = r"C:\Program Files\Tesseract-
OCR\tesseract.exe"
def
    preprocess_image(image_p
    ath): """Preprocess image
    for prediction."""
     img = load_img(image_path,
    target_size=(224, 224)) img_array =
     img_to_array(img)
```

```
    img_array
    =np.expand_dims(img_array, axis=0)
    img_array /= 255.0
    return img_array

def extract_text_from_image(image_path):
    """Extract text from an image using
    Tesseract OCR.""" img =
    Image.open(image_path)
    extracted_text =
    pytesseract.image_to_string(img)
    return extracted_text.strip()

# --------- Text Classification Setup ---------
# Define the
CustomXLMGRUTransformer
class class
CustomXLMGRUTransformer(nn.
Module):
    def _init_(self, hidden_size=512, num_labels=2, dropout_rate=0.5):
        super(CustomXLMGRUTransformer, self)._init_()
        self.xlm_roberta = XLMRobertaModel.from_pretrained('xlm-roberta-base')

        # BiGRU layer
        self.gru = nn.GRU(input_size=768, hidden_size=hidden_size,
        batch_first=True,
bidirectional=True)

        # Transformer layer
        self.transformer_layer = nn.TransformerEncoderLayer(d_model=hidden_size
        * 2, nhead=8)

        # Batch normalization
        self.batch_norm = nn.BatchNorm1d(hidden_size * 2)

        # Dropout layer
        self.dropout = nn.Dropout(dropout_rate)

        # Fully connected layers
        self.fc1 =
        nn.Linear(hidden_size * 2,
        hidden_size) self.fc2 =
        nn.Linear(hidden_size,
        num_labels)
```

```
    def forward(self, input_ids_xlm, attention_mask_xlm): xlm_outputs
        = self.xlm_roberta(input_ids_xlm,
        attention_mask=attention_mask_xlm) sequence_output =
        xlm_outputs.last_hidden_state

        gru_outputs, _ = self.gru(sequence_output)
        transformer_output =
        self.transformer_layer(gru_outputs)
        pooled_output =
        torch.mean(transformer_output,
        dim=1) normalized_output =
        self.batch_norm(pooled_output)

        combined =
        self.dropout(normalized
        _output) combined =
        torch.relu(self.fc1(comb
        ined)) logits =
        self.fc2(combined)
        return logits

# Load text classification model
TEXT_MODEL_PATH = 'FINAL
XLM.pth' text_model =
CustomXLMGRUTransformer()
text_model = torch.load(TEXT_MODEL_PATH,
map_location=torch.device('cpu')) text_model.eval()

# Load tokenizer
tokenizer = XLMRobertaTokenizer.from_pretrained('xlm-roberta-base')

def classify_text(text):
    """Classify text as Bully or Non-Bully.""" inputs = tokenizer(text,
    return_tensors="pt", truncation=True, padding=True, max_length=50)
    with torch.no_grad(): logits = text_model(inputs['input_ids'],
    inputs['attention_mask']) predicted_class = torch.argmax(logits,
    dim=1).item()
    return "Bully" if predicted_class == 1 else "Non-Bully"

# --------- Flask Routes ---------
@app.route('/')
def home():
return
```

```
render_template('
home.html')

@app.route('/about')
def about():
return
render_template('
about.html')

@app.route('/prediction')
def prediction_page():
return
render_template('predi
ction.html')

@app.route('/metrics')
def metrics():
return
render_template('m
etrics.html')

@app.route('/flowc
hart,)
def flowchart():
    return render_template('flowchart.html')

@app.route('/predict',
methods=['POST']) def
predict():
    """Handle image upload, prediction, and
    text classification.""" if 'file' not in
    request.files:
        return jsonify({'error': 'No file
    uploaded'}), 400 file =
    request.files['file']
    if file.filename == '': return
        jsonify({'error': 'No file
        selected'}), 400 filename
        =
        secure_filename(file.filen
        ame) filepath =
        os.path.join('uploads',
        filename)
        os.makedirs('uploads',
```

```
        exist_ok=True)
    file.save(filepath)

    # Image classification
    img_array =
    preprocess_image(filepat
    h) prediction =
    image_model.predict(im
    g_array)
    image_label = 'Non-Bully' if prediction[0] >= 0.5 else 'Bully'

    # Text extraction
    extracted_text = extract_text_from_image(filepath)

    # Text classification if
    extracted_text:
    text_label =
    classify_text(extracted
    _text)
    else:
        text_label = "No text extracted"

    # Clean up uploaded file
    os.remove(filepath)

    if(image_label == 'Bully' or
        text_label == 'Bully'): return
        jsonify({'prediction': 'Bully'})
    else:
        return jsonify({'prediction': 'Non-Bully'});


    #return jsonify({'image_prediction': image_label, 'extracted_text': extracted_text,
'text_prediction': text_label})


    #return jsonify({'image_prediction': image_label, 'text_prediction': text_label})

if _name ___== '_main_':
    app.run(debug=True)
```

9. RESULT ANALYSIS

9.1 Accuracy Comparison on Image Data

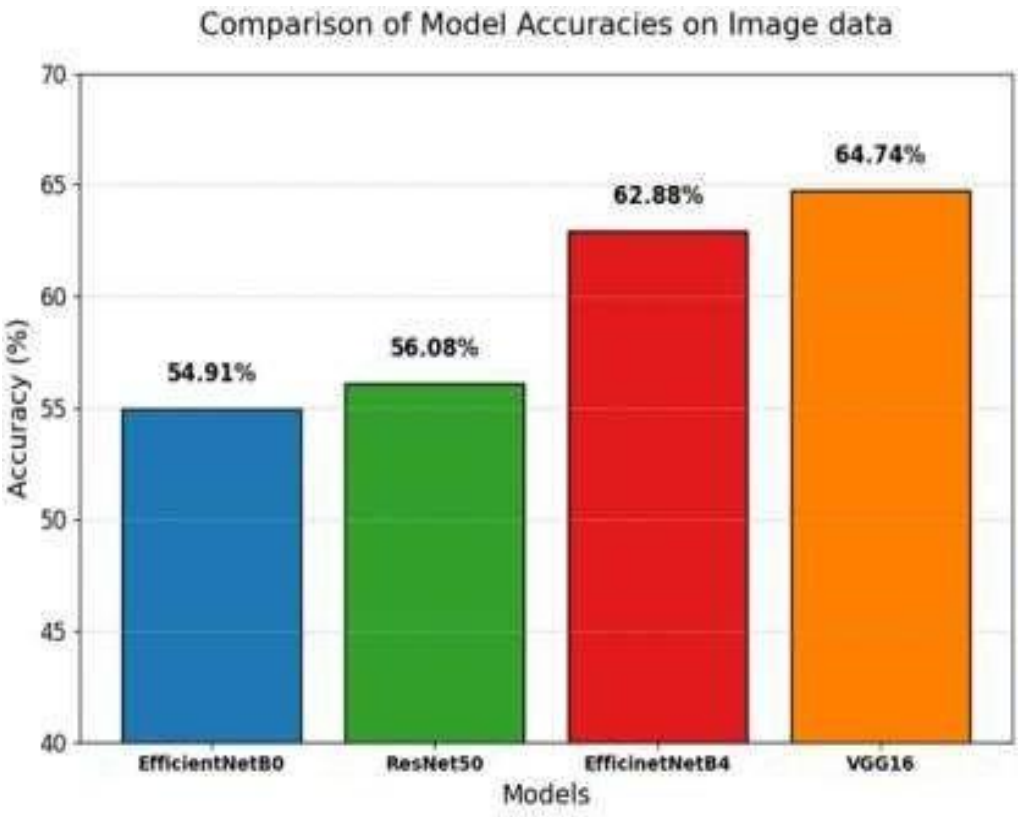

Fig 9.1 Accuracy Comparison on Image Data

The figure 9.1 illustrates the performance of four prominent deep learning architectures: EfficientNetB0, ResNet50, EfficientNetB4, and VGG16. Each model was trained and tested on the preprocessed image dataset comprising meme images labeled as bullying or non-bullying.

9.2 Accuracy Comparison on Text Data

Figure 9.2 presents the comparison of model accuracies on text data. We evaluated four different combinations of language models and neural network architectures: mBERT, XLM-RoBERTa, mBERT + BiGRU, and XLM-RoBERTa + BiGRU. These models were trained on a dataset consisting of text captions extracted from cyberbullying-related posts.

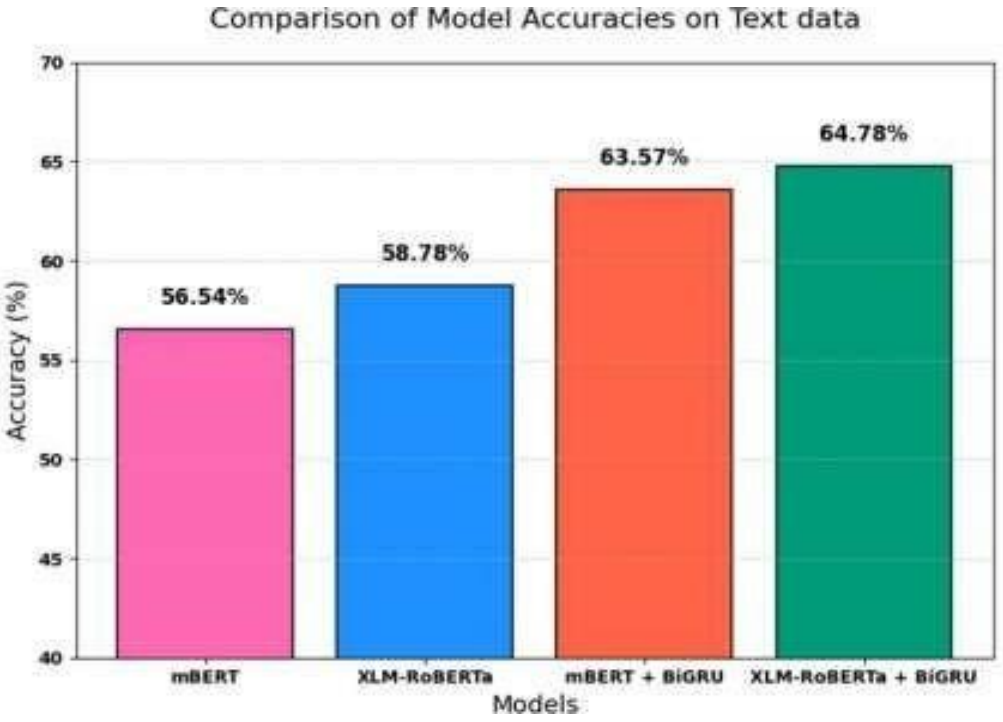

Fig 9.2 Accuracy Comparison on Text Data

9.3 ROC Curve Analysis

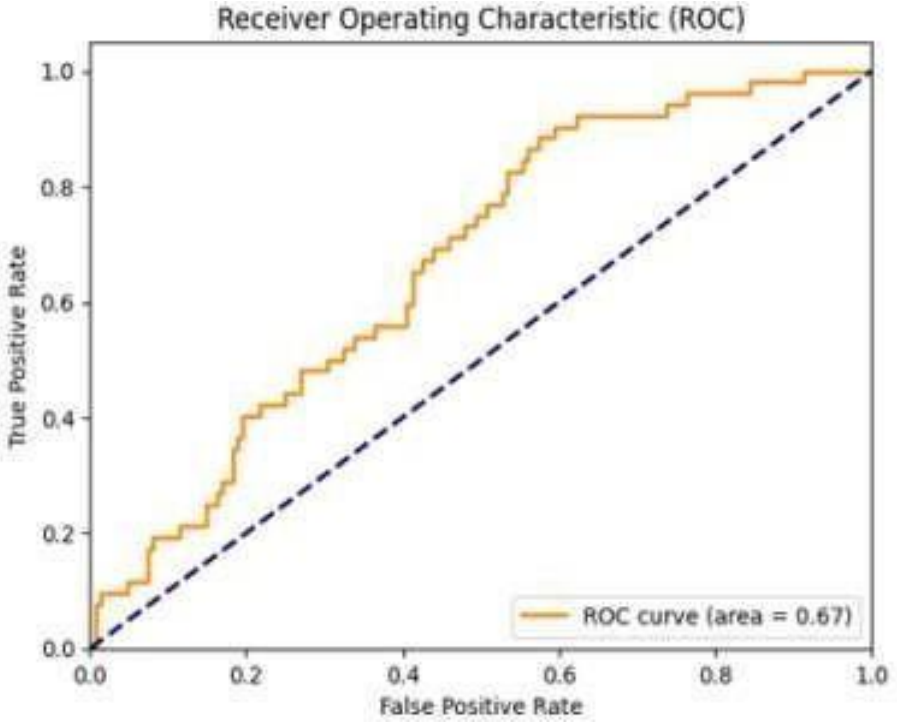

Fig 9.3 ROC curve

Figure 9.3 displays the Receiver Operating Characteristic (ROC) curve for our

multimodal cyberbullying detection model. The ROC curve plots the True Positive Rate (Sensitivity) against the False Positive Rate (1 - Specificity), providing a graphical representation of the model's performance across different classification thresholds.

9.4 Integration of Best Models in Multimodal Framework

Following the individual evaluations of image and text models which are mentioned in Table 9.1 , we integrated the best-performing models, VGG16 for image data and XLM-RoBERTa + BiGRU for text data, into a multimodal framework. The integration employed multiple fusion strategies, including Intermodal Attention Mechanisms, Fully Connected Layers, CLIP Projectors, Feedback Mechanisms, and CentralNet architectures.

Research	**Methodology**	**Accuracy**	**Remarks**
Singh et al. (2021)	CNN + LSTM	68%	No intermodal fusion; limited to memes.
Patel et al. (2022)	BERT + ResNet50	71%	Used late fusion, limited attention integration.
This Work (2025)	VGG16 + XLM-RoBERTa + BiGRU + Fusion	74%	Advanced fusion and attention mechanisms.

Table 9.1 Best Models in Multimodal Framework

9.5 Confusion Matrix

Figure 9.4 and Figure 9.5 displays the model’s performance in classifying cyberbullying content. It shows that 328 bully posts and 279 non-bully posts were correctly classified, while 311 bully posts were misclassified as non-bully and 242 non-bully posts were misclassified as bully. The high misclassification rate suggests the need for further optimization, possibly through improved feature extraction or balancing techniques.

The results of Multimodal indicate that our proposed model achieves a decent

accuracy of 74% in detecting cyberbullying in multimodal data.

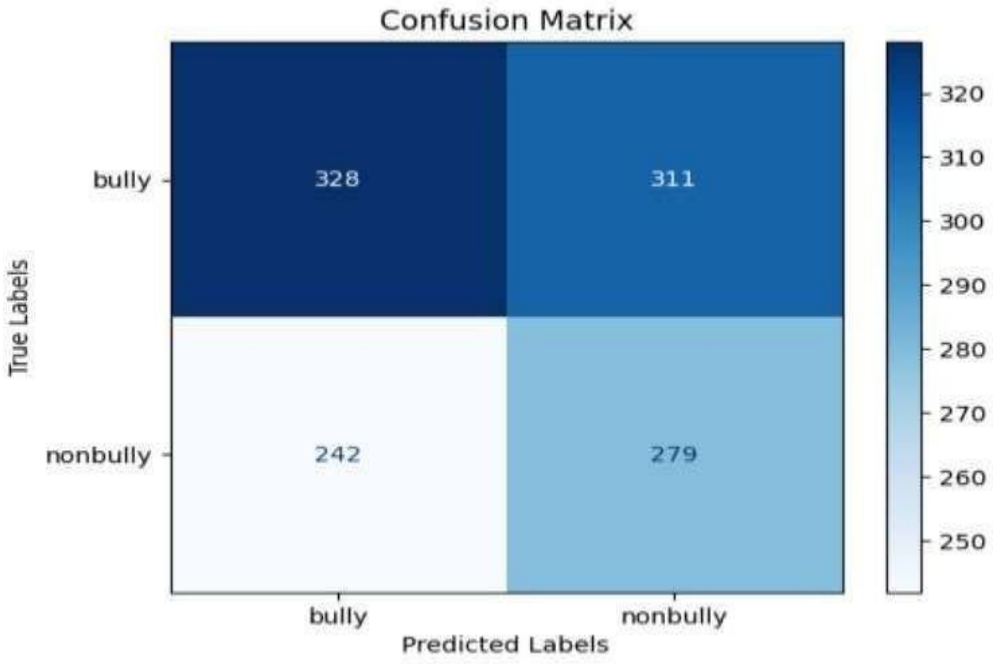

Fig 9.4 Confusion Matrix of Image Model

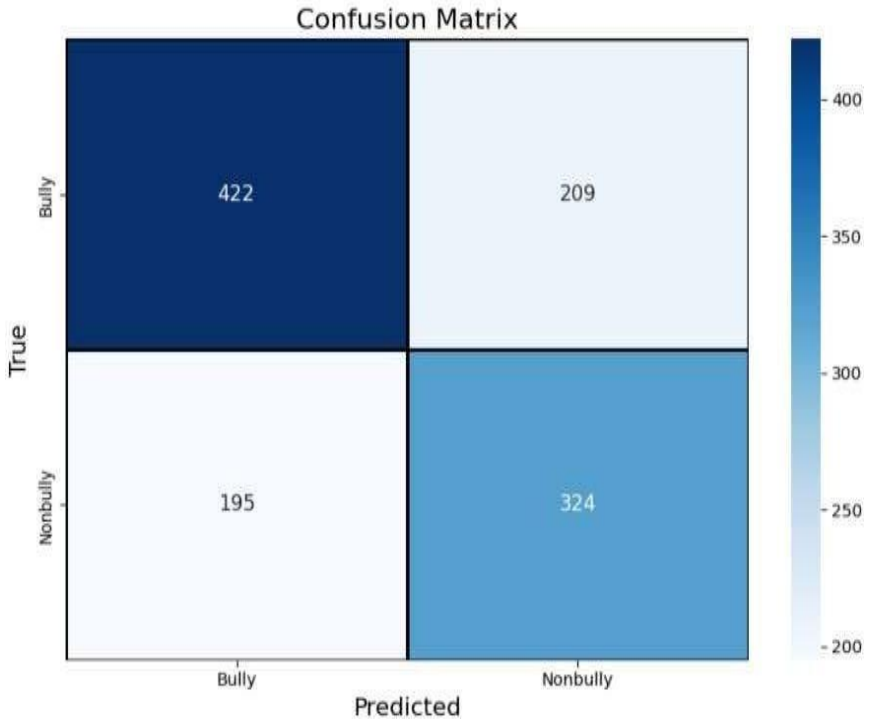

Fig 9.5 Confusion Matrix of Text Model

10. TESTCASES

Fig 10.1 Non-Bully

After uploading an image the classifier detects it as Non-Bully as shown in Fig 10.1

Fig 10.2 Bully

After uploading an image the classifier detects it as Bully as shown in Fig 10.2

11. USER INTERFACE

Fig 11.1 Home Screen

Fig 11.1 is Home Screen which shows the main web page that a visitor will view when they navigate to our website.

Fig 11.2 About Screen

Fig 11.2 is About the Project where it consists of project results and accuracy.

Fig 11.3 Prediction Screen

Fig 11.3 is Prediction Screen which helps to classify the Bully and Non-Bully content.

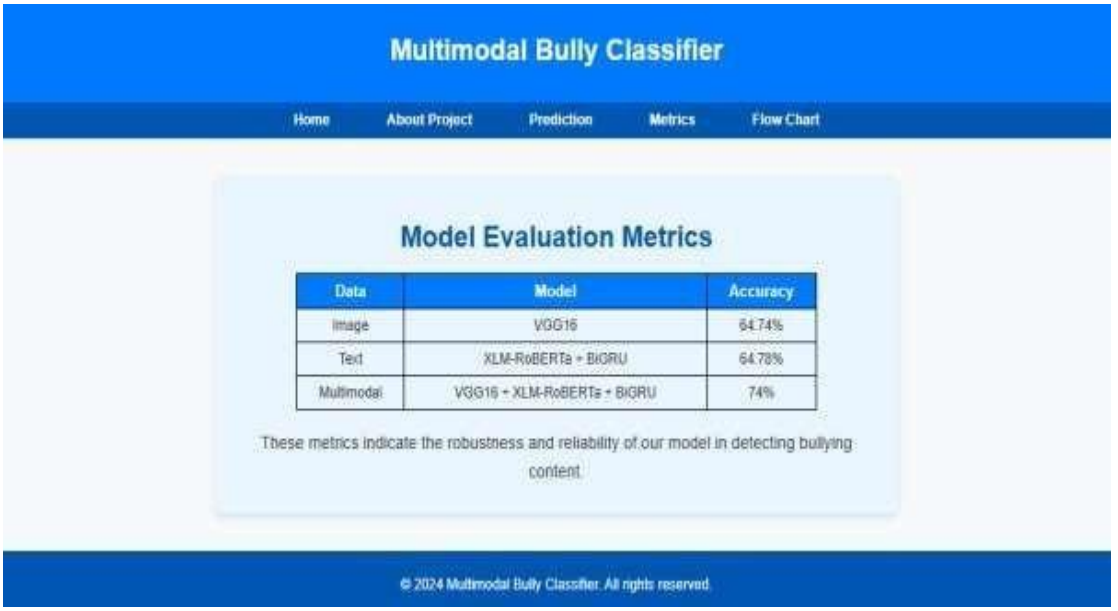

Fig 11.4 Metrics Screen

Fig 11.4 is Metrics Screen consists of Models that we used to find best accuracy.

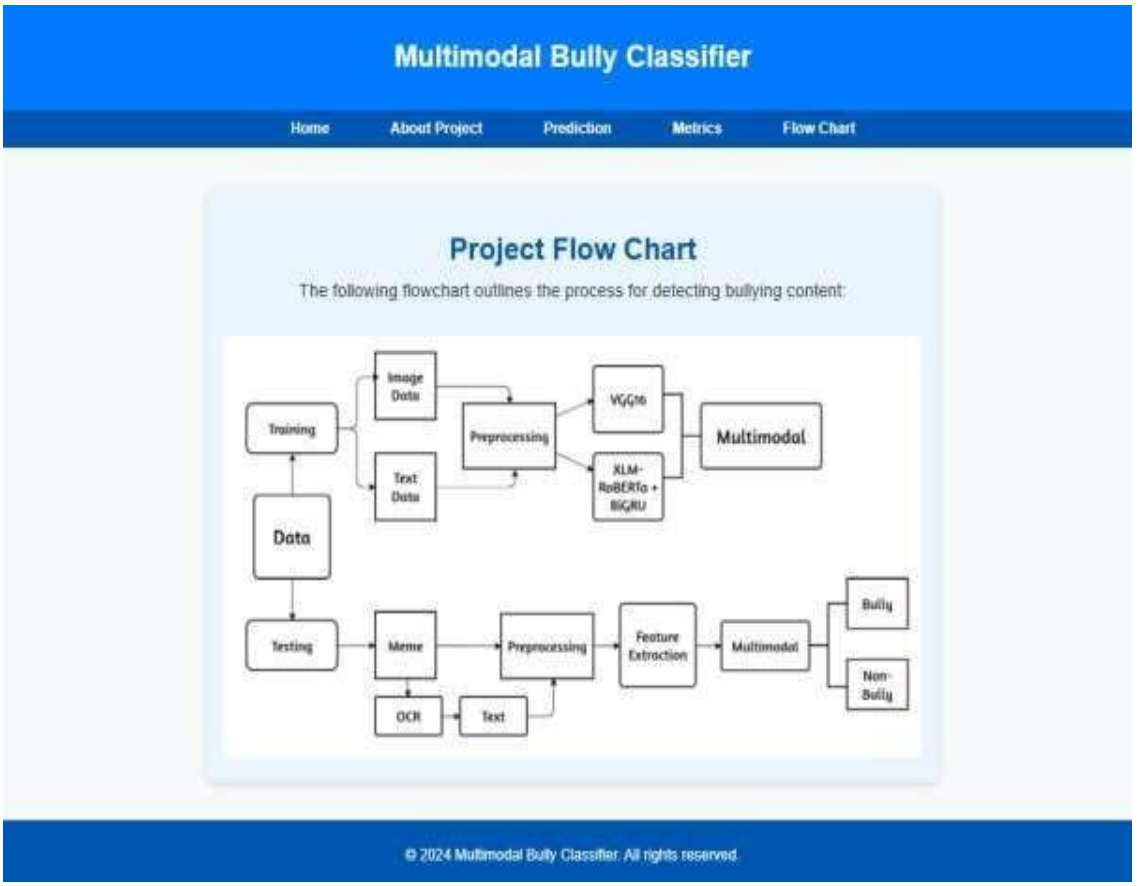

Fig 11.5 Flowchart Screen

Fig 11.5 is Flowchart Screen where it shows us whole Detection process and how Detection takes place.

12. CONCLUSION

The proposed study offers a strong and innovative approach for the detection of cyberbullying in multimodal data by combining advanced deep learning techniques for text and image analysis. Cyberbullying is one of the most widespread issues in today's digital landscape, posing significant psychological and emotional risks, and hence effective detection mechanisms are required. Unlike traditional methods that focus solely on text-based content, this research addresses the complexities of analyzing multimodal data, including memes that integrate both visual and textual components. By leveraging pre-trained VGG16 for image feature extraction and XLM-RoBERTa with BiGRU for text analysis, the model demonstrates superior capability in identifying bullying patterns. Integration of CLIP projectors and intermodal attention mechanisms would deepen the contextual understanding achieved through feature alignment and interpretation in modalities. Through rigorous data preprocessing such as augmentation, normalization, and tokenization, the dataset was cleaned to achieve consistency and quality, thus eliminating the biases that may arise. The multimodal framework results in an accuracy of 74%, which is more than that of individual text and image models. This performance proves that visual and textual cues can be combined to identify implicit and explicit bullying behaviors. The study also underlines the importance of feature fusion and feedback mechanisms for improving predictions and enhancing interpretability.

Furthermore, dropout layers, AdamW optimizer, and early stopping techniques prevent overfitting, ensuring that the model generalizes well across unseen data.

The evaluation metrics, including precision, recall, and F1-score, confirm the robustness and reliability of the approach. By addressing both text and image data, this research bridges a crucial gap in existing methodologies, offering a scalable and adaptable solution to cyberbullying detection. The findings highlight the need for multichannel approaches when combating complex online harassment patterns. This opens up scope for future improvements and wider applications. At the end, it also contributes to the safety of online environments, fosters respect, and lessens the negative effects of cyberbullying. Future Scope Although the proposed model demonstrates promising results, several avenues for future research and development remain.

13. FUTURE SCOPE

Further improvements to the model's accuracy and scalability can make it more practically applicable.

One major area of improvement is the inclusion of additional modalities, such as audio and video data, which can provide deeper insights into bullying behaviors on platforms like TikTok and YouTube. Incorporating these modalities would enable the detection of bullying in multimedia content, addressing evolving patterns in online interactions. Another possible improvement can be made in intermodal attention mechanisms to be capable of dealing with more complicated and ambiguous data. Advanced transformers like Vision Transformers (ViT) and GPT-based models can be added for better contextual understanding and semantic alignment across modalities. Besides, language-specific and culturally adaptive models can be built to address linguistic and regional variations, thus increasing the applicability of the framework to non-English datasets and diverse cultural contexts. Future work will also address real-time detection systems with the ability to respond even more quickly to cyberbullying incidents. Using edge computing techniques and lightweight neural networks enables the deployment of such mechanisms on mobile devices and platforms for social media, therefore monitoring in real-time and reporting incidents. Also, incorporating explainable AI (XAI) methods will help make models more transparent for users to understand model predictions and reduce bias. The inclusion of reinforcement learning and adversarial training techniques can make the model even more robust, capable of adapting to adversarial attacks or evolving bullying patterns. Expanding datasets to include more diverse samples with labels like sarcasm, humor, and implicit bullying will improve generalization. Interaction with psychologists and social scientists can help refine annotation procedures and address ethical concerns.

Lastly, the study may be expanded into developing intervention strategies in which the model not only identifies bullying but also recommends response mechanisms or resources to the victims. Automated reporting mechanisms and content moderation systems can help social media companies to proactively prevent bullying behaviors. The development will thus make the system comprehensive and effective and would contribute to a great deal to fighting cyberbullying.

14. REFERENCES

1. Debnath, Dipanwita & Das, Ranjita & Rafi, Shaik. (2022). Sentiment-Based Abstractive Text Summarization Using Attention Oriented LSTM Model. 10.1007/978- 981-16-6624- 7_20.

2. Rafi, S., Das, R. Topic-guided abstractive multimodal summariza- tion with multimodal output. Neural Comput & Applic (2023). https://doi.org/10.1007/s00521-023-08821-5

3. Greeshma, B., Sireesha, M., Thirumala Rao, S.N. (2022). Detection of Arrhythmia Using Convolutional Neural Networks. In: Shakya, S., Du, KL., Haoxiang, W. (eds) Proceedings of Second International Conference on Sustainable Expert Systems . Lecture Notes in Networks and Systems, vol 351. Springer, Singapore. https://doi.org/10.1007/978-981-16-7657-4_3

4. S. L. Jagannadham, K. L. Nadh and M. Sireesha, "Brain Tumour Detection Using CNN," 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2021, pp. 734-739, doi: 10.1109/ISMAC52330.2021.9640875

5. A. Varsha Reddy, Gugulothu Kalpana, N. Satish Kumar, and Dr. Sundaragiri Dheeraj, “Cyber Bullying Text Detection Using Machine Learning”, June 2022 , Available: https://doi.org/10.22214/ijraset.2022.44157

6. Alabdulwahab, Aljwharah & Haq, Mohd Anul & Alshehri, Mohammed. (2023). Cyberbullying Detection using Machine Learning and Deep Learning. International Journal of Advanced Computer Science and Applications. 14. 424-432. 10.14569/IJACSA.2023.0141045.

7. Pradeep, Kumar, Roy., Fenish, Umeshbhai, Mali. (2022). Cyberbullying detection using deep transfer learning. Complex & Intelligent Systems, 8(6):5449-5467. doi: 10.1007/s40747- 022-00772-z

8. Vishwamitra, Nishant & Hu, Hongxin & Luo, Feng & Cheng, Long. (2021). Towards Understanding and Detecting Cyberbullying in Real-world Images. 10.14722/ndss.2021.24260.

9. Pericherla, Subbaraju & Ilavarasan, E.. (2024). Overcoming the Chal- lenge of Cyberbullying Detection in Images: A Deep Learning Approach with Image

Captioning and OCR Integration. International Journal of Computing and Digital Systems. 15. 393-401. 10.12785/ijcds/150130.

10. Kumari, Kirti & Singh, Jyoti & Dwivedi, Yogesh & Rana, Nripendra. (2020). Towards Cyberbullying-free social media in smart cities: a unified multi- modal approach. Soft Computing. 24. 10.1007/s00500- 019-04550-x

11. Wang, Kaige & Xiong, Qingyu & Wu, Chao & Gao, Min & Yu, Yang. (2020). Multi-modal cyberbullying detection on social networks. 1-8. 10.1109/IJCNN48605.2020.9206663.

12. Ahmed, Md.Tofael & Akter, Nahida & Rahman, Maqsudur & Islam, Abu & Das, Dipankar & Rashed, Md. Golam. (2023). Multimodal Cy- berbullying Meme Detection From Social Media Using Deep Learning Approach. International Journal of Computer Science and Information Technology. 15. 27-37. 10.5121/ijcsit.2023.15403.

13. Maity, Krishanu & Saha, Sriparna & Bhattacharyya, Pushpak. (2022). A Multitask Framework for Sentiment, Emotion and Sarcasm aware Cyberbullying Detection from Multi- modal Code-Mixed Memes. 1739- 1749. 10.1145/3477495.3531925.

Multimodal Cyberbullying Detection Using Deep Learning Techniques

Shaik Rafi[1], Ranjita Das[2], Kalyanam Jahnavi Sai Priya[3], Bolla Lakshmi Varsha[3], Velchuri Bala Harshitha[3], Sunkari Kavya[3], T.G. Ramnadh babu[4], and Sireesha Moturi[5]

[1] Asst.Professor, Dept of Computer Science and Engineering
Narasaraopeta Engineering College (Autonomous),
Narasaraopet 522601, Palnadu District, Andhra Pradesh, India.
shaikrafinrt@gmail.com
[2] Asst.Professor, Dept of Computer Science and Engineering
National Institute of Technology Agartala,
Barjala-799046, Jirania, West Tripura, Agartala, Tripura, India.
ranjita.nitm@gmail.com
[3] Dept of Computer Science and Engineering
Narasaraopeta Engineering College (Autonomous),
Narasaraopet 522601, Palnadu District, Andhra Pradesh, India.
kjahnavisaipriya@gmail.com, varshibolla2507@gmail.com,
vharshitha738@gmail.com, kavyasunkari510@gmail.com
[4] Asst.Professor, Dept of Computer Science and Engineering
Narasaraopeta Engineering College (Autonomous),
Narasaraopet 522601, Palnadu District, Andhra Pradesh, India.
baburamnadh@gmail.com
[5] Assoc.Professor, Dept of Computer Science and Engineering
Narasaraopeta Engineering College (Autonomous),
Narasaraopet 522601, Palnadu District, Andhra Pradesh, India.
sireeshamoturi@gmail.com

Abstract. Cyberbullying detection is the process of classifying and rec- ognizing cyberbullying activity, which includes using technology to ha- rass or threaten people—usually via online platforms. In order to address this, we examined a publicly available dataset that was classified as bully or non-bully according to text, image and image-text. We next proposed applying a deep learning model to recognize cyberbullying based on mul- timodal data. The VGG16 pre-trained model detects bullying in photos, while the XLM-RoBERTa with BiGRU model detects bullying in text. By combining these models (VGG16 + XLM-RoBERTa and BiGRU) with attention processes, CLIP, feedback mechanisms, CentralNet and other tools, we created a model for detecting cyberbullying in image- text based memes. Our final model showed that the algorithm is able to identify most cyberbullying occurrences with a decent accuracy of 74%.

Keywords: Cyberbully · VGG16 · XLM RoBERTa · BiGRU.

1 Introduction

In the cutting-edge technology, social media platforms and virtual era are getting used by number of people. Numerous services, along with WhatsApp, Instagram, Facebook and Twitter, are liable to cyberbullying. Cyberbullying is the use of digital era to annoy, harass, or threaten someone. Recognizing and responding to cyberbullying is vital to strengthen the protection against online bullying.

Bullying through images and text [1] are two forms of cyberbullying. The act of sending threatening, intimidating messages to another person via social media, messaging, or email is known as text-based bullying. The consequences of such an act can result in serious psychological damage and emotional distress. Image bullying occurs when someone posts offensive images or improper images in online without permission. It can be very embarrassing and very damaging to a person's reputation and sense of worth. Both of these bullies can have negative consequences and that's the point. So we need to work together to prevent and end online bullying, especially when it comes to young people.We have to establish a welcoming, respectful and secure on-line network for everyone.

Since bulk of modern studies focused on text-based situations, image-based cyberbullying has not got attention as text-based cyberbullying. This is an ex- cessive issue as several messages on social media contain both text and images. To address this, we are going to buid a multi-modal [2] that is capable of de- tecting bullying in both images and texts. Our main goal is to create a pleasant and secure online environment where each person may thrive with out fear of abuse or intimidation. Even though it is a challenging endeavor, our intention is to make internet users lives better.

With the use of many modalities, our research attempts to detect cyberbul- lying in image-text posts. To extract the necessary attributes from images, deep learning models use convolutional neural networks (CNNs) [3,4] and transfer learning techniques. The study additionally employs Ensemble XLM-RoBERTa with BiGRU and VGG16, to enhance the detection of text and image based cyberbullying on social media platforms.

The rest of the paper is organized as follows: Literature Survey is discussed in Section 2 and Section 3 presents the Methodology. Whereas Section 4 highlights about Experimental Setup, Results are displayed in Section 5 and conclusion in Section 6.

2 Literature Survey

The growing number of social media platforms resulted to a corresponding rise in cyberbullying, prompting numerous studies to focus on efforts aimed at detecting cyberbullying. Some research on the detection of cyberbullying is discussed in this section.

2.1 Works on Text Data

A model that uses a machine learning algorithm to identify cyberbullying in literature was created by Varsha Reddy et al. [5]. They gathered postings and comments from other social media sites, including Facebook and Twitter. With an accuracy of 80.01%, the Logistic Regression model outperformed all other tested models. Using a dataset of 47,692 tweets, Mohammad Alshehri et al. [6] created a model utilizing CNN and LSTM. Their proposed model's accuracy of 96% demonstrates its effective ability to identify cyberbullying.

2.2 Works on Image Data

Using deep transfer learning models, Pradeep Kumar Roy et al. [7] created a model for the detection of cyberbullying in images. The 3000 image dataset was created by hand, with 1458 photos classified as bullies and 1542 images classified as non-bullies. They employed many deep transfer learning models, and InceptionV3 performed well, obtaining an accuracy rate of 89%. Using a combination of methods, Nishant Vishwamitra et al. [8] employed VGG16 and MLP to create a model that can identify cyberbullying in images. After 19,300 photos were used to train the model, its accuracy exceeded 93.36%.

2.3 Works on Multimodal Data

Using data from Facebook, Instagram and Twitter, Subbaraju Pericherla et al. [9] created a model that focuses on visuals and text features in social media images. They used OCR, VGG16 and LSTM to extract text and BEiT and MLP were used to create a CNBD model that had an accuracy of 98.23%. The CNN-based multi-model deep learning system developed by Kumari et al. [10] achieved 71% accuracy for bullying posts and 64% accuracy for non-bullying posts. They produced a dataset comprising 2100 posts, of which 1418 were deemed to be bullying and 619 to not be. While most studies focus on text and graphics, multimodal material also includes photos, videos and audio.

The Multi-Modal Cyberbullying Detection (MMCD) framework was introduced by Quingyu Xiong et al. [11]. It employs models such as TF-IDF, HAN, BiLSTM and Visual Embedding to detect cyberbullying content. Using Vine data, the machine obtained 83% accuracy. 86% accuracy rate on Instagram data shows that it can identify cyberbullying. The ability to identify cyberbullying content over a wide linguistic spectrum has improved with research. Using VGG16 and BiLSTM, Nahida Akter et al. [12] developed a hybrid deep neural network model to detect cyberbullying content in Bengali memes. With an accuracy of 87%, their investigation produced the Bengali Memes Dataset, which consists of 600 memes associated with bullying and 600 unrelated memes.

3 Methodology

This section covers preprocessing methods and models for text and image data. Additionally, we demonstrated our approach for detecting cyberbullying in mul- timodal data and the hyperparameters that the multimodel uses.

3.1 Data Pre-processing

Image Pre-processing The pre-processing techniques applied to the image data are as follows. These pre-processing methods helps in enhancing the data, which raises the model's performance.

- **PIL Library :**
  For opening, editing, and storing images as well as for image enhancements, the Python Imaging Library (PIL) is used for a variety of image-related tasks. It assists in locating and removing improperly formatted images.
- **Data Augmentation and Normalization (Using Image-DataGen- erator) :**
  Data augmentation can improve model performance by expanding the size of the data. This is accomplished by using ImageDataGenerator class meth- ods like brightness range, rotation range, zoom range, horizontal flip and width shift range etc. Pixel values must be normalized for deep learning models. This entails using a 1./255 rescaling factor to convert the pixel val- ues from [0, 255] to [0, 1]. Specifically, this ensures data consistency with model predictions whether transfer learning or pre-trained models are used. The augmented image is shown in Fig. 1.

Fig. 1: Image Augmentation

- **Image Resizing :**
  Images are scaled to a specified objective dimension, such as (224, 224), be- fore being fed into the neural network to ensure input dimension consistency for pre-trained models such as VGG16. This step is crucial for models like VGG16, which might not work as well with images of varied sizes.

Text Pre-processing Pre-processing techniques are necessary for text gener- ation models to ensure that the data is suitable for deep learning applications. The steps in this process are loading, cleaning, entering the data into models and these are covered in more detail below.

- **Data Loading and Cleaning :**
  We import the dataset using the pandas package's pd.read csv function. As part of data cleansing, duplicate values, null values, and superfluous columns must be eliminated. It is possible to re-move unwanted columns with df.drop(). df.dropna() can be used to remove rows that have miss- ing values, preventing incorrect predictions from being made as a result of incomplete data. Duplicate rows can result in overfitting, hence df.drop du- plicates() aims to remove them. After pre-processing, the sample size was reduced from 5865 texts to 5749 texts, of which 3149 texts were bullies and 2600 texts were not.
- **Label Encoding :**
  Label encoding is crucial when handling classification problems involving cat- egorical features. The deep learning method converts labels into a numeric format, such as 0 and 1, to make them simpler for the system to under- stand.This is done by using the LabelEncoder function of the scikit-learn toolkit.
- **Tokenization (Text Preprocessing) :**
  Tokenization is a technique used by deep learning models to convert raw text into numerical representations. The XLMRober-taTokenizer splits input text into subword units to make code-switching between languages or managing complicated morphological languages easier. After tokenizing the text, it may now be used for training and testing models.
- **Train-Test Split :**
  To assess a model's performance, the data is divided into train and test sets using a predetermined procedure. This separation helps in evaluating the performance of the modal accurately.
- **Label Conversion to Tensor :**
  Labels are converted to PyTorch tensors before training the model, much like tokenized text conversion, to make sure they are in the correct format.

3.2 Image Based detection

In order to identify cyberbullying in photographs, we have experimented with a number of deep learning models on the image data. We trained the images using

ResNet50, EfficientNetB0, EfficientNetB4, and VGG16, using various hyperpa- rameters. Fig. 2 shows comparisons and the accuracy of each model. Using our data, the VGG16 pre-trained model performed better than any other model. We

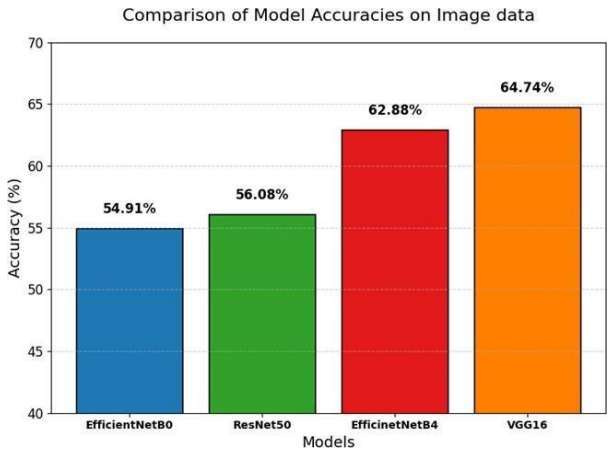

Fig. 2: Comparisons Of Image Models

employed a range of hyperparameters, such as the Adam optimizer, Dropout, Global Average Pooling, Dense layers etc., to optimize the training. Fig. 3 depicts the process clearly. After training the model across several epochs, we have also optimized it using a lower learning rate for better results.

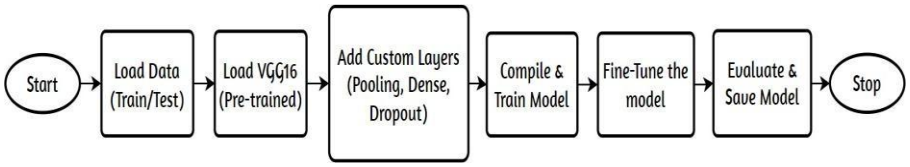

Fig. 3: Flow diagram of Image Modal

3.3 Text Based detection

Similar to the image training, we trained several models on the text data before deciding on the best-performing model. The models that we employed are mBERT, mBERT with BiGRU, XLM-RoBERTa, and XLM-RoBERTa paired

with BiGRU. When all of these models are compared, Fig. 4 demonstrates that the XLM-RoBERTa ensemble with BiGRU had the best performance.

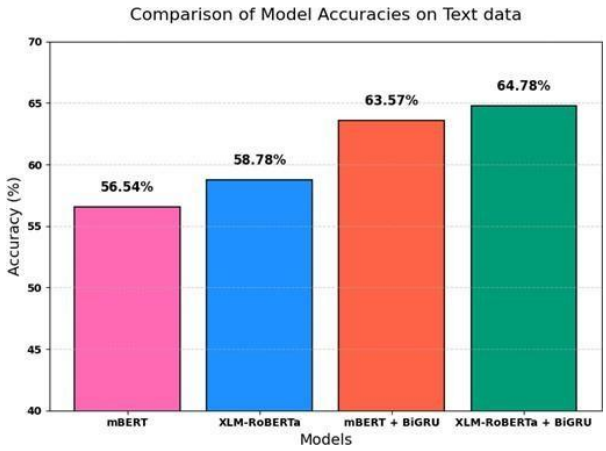

Fig. 4: Comparisons Of Text Models

The XLM-Roberta language model is integrated with a special design that includes a transformer layer to help with context understanding and a special kind of memory layer (bidirectional GRU), as shown in Fig. 5. We used a number of novel strategies, including batch normalization and dropout, to ensure that the model maintained its capacity to generalize and didn't become overly skilled at fitting the training set.In the end, we used AdamW, a smart optimizer that gradually changes the learning rate, to modify the training process and raise the model's learning efficiency. With these algorithms and architectural features combined, the model performs very well in its role, making it a dependable answer for text categorization problems.

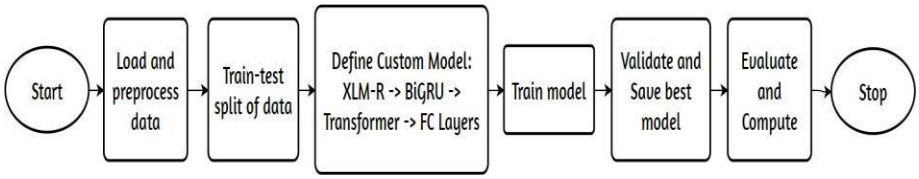

Fig. 5: Flow diagram of Text Modal

3.4 Multi-modal detection

After training the data independently, it's time to integrate the models that per- form best with the text and image data. We created a new model by combining the pre-trained VGG16 model (used for images) with the XLM-RoBERTa and BiGRU model (used for text) using a number of strategies, including Intermodal Attention, Feedback Mechanism, CentralNet, CLIP Approach, and Fully Con- nected Layers. The entire flow of the model is displayed in Fig. 6. The model was trained using 80% of the available data (text and images), and its performance was verified using additional data. Hyperparameters, which are specified prior to the model's training phase, regulate the learning process and model architec- ture. The hyper parameters used in our proposed model to identify cyberbullying content in multimodal data is shown in Table 1.

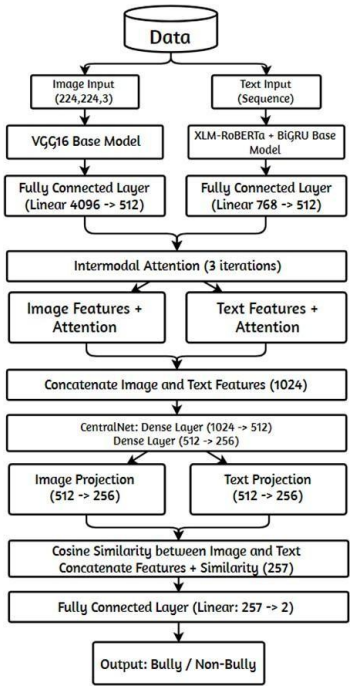

Fig. 6: Multimodal data detection

Table 1: Hyperparameters used in the model

Component	Hyperparameter	Value
Image Model (VGG16)	Dropout Rate	0.5
	Fully Connected Layer	512 (from 25088)
Text Model (XLM-RoBERTa)	Fully Connected Layer Output Size	512 (from 768)
	Image Projector	256 (from 512)
CLIP projectors	Image Projector	256 (from 512)
	Text Projector	256 (from 512)
Intermodal Attention	Embedding Dimension	512
	Number of Attention Heads	8
Final Classifier	Output Size	257 (256 + 1)
	Number of Classes	2
Training Settings	Batch Size	32
	Learning Rate	1e-5
	Weight Decay	0.01
	Early Stopping Patience	3
	Loss Function	Cross Entropy
	Optimizer	AdamW

4 Experimental Setup

4.1 Dataset Description

The code-mixed text and image posts from Sriparna Saha et al.'s [13] 2022 dataset served as the basis for our investigation. The dataset used in this study is

publicly available and can be accessed here for the images - https://drive.google.com/drive/folders/1vCgjNvgVFDl3SVsGmyxx3urJOgYeLIjs and here for the text data - https://docs.google.com/spreadsheets/d/1tD5yqGZ3TlDjeUFThautfZGegHrRz7FW/edit?gid=1650123160#gid=1650123160. A wide range of topics are covered by the dataset, such as harm-fulness, emotion, sarcasm, and cyberbully- ing. Since we are dealing with the detection of cyberbullying, we only choose the cyberbullying section. Among those the crucial columns are image name, text, image label, image-text label, and image-text label. More complicated models can be developed by combining textual and visual data to improve categoriza- tion accuracy and robustness. The dataset contains 6,006 posts in total, which are separated into bully and non-bully categories. Because there are so many images, a detailed analysis of bullying and non-bullying is possible. The dataset has been reduced to 5798 by pre-processing, with 3,193 bully images and 2,605 non-bully images. This made it possible to distribute data fairly and train and evaluate models in an effective manner.

4.2 Data Preparation

Images Data To achieve strong performance metrics and avoid over-fitting, the image-based model has an 80-20 split between the train and test parts. Of all the images, 20% are in the test directory (1160) and 80% are in the train directory (4638). The train directory further splits images into bully and non-bully classes to provide clear class identification. Out of the 4638 photos, 2,554 are associated with the bully class and 2,084 with the non-bully class. The test directory does not need to be further divided because we just utilize it for testing and validation. To get information, however, we counted the number of images that included and excluded bullies.The test directory has 1160 images total—639 images of bullies and 521 images of non-bully. Table 2 displays the data's overall distributions.

Table 2: Distribution of images

Split	Images	Bully	Non Bully
Train	4638	2554	2084
Test	1160	639	521
Total	**5798**	**3193**	**2605**

Text Data The dataset we examined had annotations like sentiment and sar-casm. In order to make cyberbullying the main topic, we removed unneeded

columns and selected text-related columns. The Img_Text and Img_Text_Label columns were chosen in the dataset. Img_Text is the text that was used to train the model, and Img_Text_Label is the class of the text that has been classified as either a bully or a nonbully. Furthermore, nearly equal percentages of texts in the textual data are texts that are bully and texts that are non-bully, which helps the model fit both kinds of data.

5 Results

We evaluated many models on text and image data and selected models with the best performance individually. We picked these best models and coupled them with a range of techniques, including feedback mechanisms, fully linked layers, CLIP projectors, Intermodal Attention, and CentralNet, to produce a modal that fits on multimodal data. More than a thousand instances are used to train this modal. After evaluating our model using test data, we obtained an overall accuracy of 74%. The model accuracy for each set of data is shown in Table 3.

Table 3: Overall Results

Data	**Model**	**Accuracy**
Image	VGG16	64.74%
Text	XLM-Roberta + BiGRU	64.78%
Multimodal (Image + Text)	VGG16 + XLM-Roberta + BiGRU	74%

6 Conclusion

In this study, we introduced a model that can identify bullying in multimodal data—that is, both text and images. We used a publicly accessible multimodal memes dataset for this, which contains images with text and these are labeled as either bully or non-bully. After experimenting with many models, we discovered that the models that perform the best on images and text are VGG16 and XLM-RoBERTa + BiGRU. Finally, we combined these models with other deep learning techniques, such as CentralNet, CLIP, Intermodal Attention, Feedback mechanism, etc. for multimodal detection, which improved our model's ability to identify bullying in multimodal data.

References

1. Debnath, Dipanwita & Das, Ranjita & Rafi, Shaik. (2022). Sentiment-Based Abstractive Text Summarization Using Attention Oriented LSTM Model. 10.1007/978- 981-16-6624-7_20.
2. Rafi, S., Das, R. Topic-guided abstractive multimodal summariza- tion with multimodal output. Neural Comput & Applic (2023). https://doi.org/10.1007/s00521-023-08821-5
3. Greeshma, B., Sireesha, M., Thirumala Rao, S.N. (2022). Detection of Arrhyth- mia Using Convolutional Neural Networks. In: Shakya, S., Du, KL., Haoxiang, W. (eds) Proceedings of Second International Conference on Sustainable Expert Systems . Lecture Notes in Networks and Systems, vol 351. Springer, Singapore. https://doi.org/10.1007/978-981-16-7657-4_3
4. S. L. Jagannadham, K. L. Nadh and M. Sireesha, "Brain Tumour Detection Using CNN," 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2021, pp. 734-739, doi: 10.1109/I- SMAC52330.2021.9640875
5. A. Varsha Reddy, Gugulothu Kalpana, N. Satish Kumar, and Dr. Sundaragiri Dheeraj, “Cyber Bullying Text Detection Using Machine Learning”, June 2022 , Available: https://doi.org/10.22214/ijraset.2022.44157
6. Alabdulwahab, Aljwharah & Haq, Mohd Anul & Alshehri, Mohammed. (2023). Cyberbullying Detection using Machine Learning and Deep Learning. International Journal of Advanced Computer Science and Applications. 14. 424-432. 10.14569/IJACSA.2023.0141045.
7. Pradeep, Kumar, Roy., Fenish, Umeshbhai, Mali. (2022). Cyberbullying detection using deep transfer learning. Complex & Intelligent Systems, 8(6):5449-5467. doi: 10.1007/s40747-022-00772-z
8. Vishwamitra, Nishant & Hu, Hongxin & Luo, Feng & Cheng, Long. (2021). Towards Understanding and Detecting Cyberbullying in Real-world Images. 10.14722/ndss.2021.24260.
9. Pericherla, Subbaraju & Ilavarasan, E.. (2024). Overcoming the Chal- lenge of Cyberbullying Detection in Images: A Deep Learning Approach with Image Captioning and OCR Integration. International Journal of Computing and Digital Systems. 15. 393-401. 10.12785/ijcds/150130.
10. Kumari, Kirti & Singh, Jyoti & Dwivedi, Yogesh & Rana, Nripendra. (2020). Towards Cyberbullying-free social media in smart cities: a unified multi-modal approach. Soft Computing. 24. 10.1007/s00500- 019-04550-x
11. Wang, Kaige & Xiong, Qingyu & Wu, Chao & Gao, Min & Yu, Yang. (2020). Multi-modal cyberbullying detection on social networks. 1-8. 10.1109/IJCNN48605.2020.9206663.
12. Ahmed, Md.Tofael & Akter, Nahida & Rahman, Maqsudur & Islam, Abu & Das, Dipankar & Rashed, Md. Golam. (2023). Multimodal Cy- berbullying Meme Detection From Social Media Using Deep Learning Approach. International Journal of Computer Science and Information Technology. 15. 27-37. 10.5121/ijcsit.2023.15403.
13. Maity, Krishanu & Saha, Sriparna & Bhattacharyya, Pushpak. (2022). A Multitask Framework for Sentiment, Emotion and Sarcasm aware Cyberbullying Detection from Multi-modal Code-Mixed Memes. 1739- 1749. 10.1145/3477495.3531925.

Plagiarism Report

ORIGINALITY REPORT

3%	3%	3%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ijream.org Internet Source	2%
2	Soukaina Bouarourou, El habib Nfaoui, Abdelhak Boulalaam, Abderrahim Zannou. "A Predictive Model for Abnormal Conditions in Smart Farming using IoT Sensors", Procedia Computer Science, 2023 Publication	1%
3	www.ncbi.nlm.nih.gov Internet Source	<1%
4	S V N Sreenivasu, Sakshi Gupta, Ghanshyam Vatsa, Anurag Shrivastava, Swati Vashisht, Aparna Srivastava. "Carbohydrate Recommendation for Type-1 Diabetics Patient Using Machine Learning", 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), 2022 Publication	<1%

Certificate

CERTIFICATE

OF PRESENTATION

AMRIT 2024

Paper ID: 115

Paper Title: Multimodal Cyberbullying Detection Using Deep Learning Techniques

Authors: Kalyanam Jahnavi Sai Priya, Bolla Lakshmi Varsha, Velchuri Bala Harshitha, Sunkari Kavya, Shaik Rafi

This is to certify that **KALYANAM JAHNAVI SAI PRIYA** *of* ***Narasaraopeta Engineering College*** *has presented the paper at the* ***Second International Conference on Advanced Computing, Machine Learning, Robotics and Internet Technologies (AMRIT-2024)****, organised in hybrid mode by the Department of Computer Science & Engineering and Department of Electrical Engineering, National Institute of Technology Agartala, Tripura, jointly organised by the Department of Computer Science, Assam University, Silchar, Assam, during November 8-9, 2024.*

Dr. Prodipto Das
Program Chair, AMRIT-2024
Dept. of CS, AU, Silchar

Dr. Sadhan Gope
Organizing Chair, AMRIT-2024
Dept. of EE, NIT Agartala

Dr. Ranjita Das
Organizing Chair, AMRIT-2024
Dept. of CSE, NIT Agartala