

# *Advanced Cyberbullying Detection using modern bert*

## **PRESENTED BY**

Dhanush Gowda HM (4PS21IS015)

Hemashree SS (4PS21IS021)

Sumanth M (4PS21IS053)

Vaishnavi C (4PS21IS056)

## **GUIDED BY**

Suresh MR

Associate Professor

Dept. of IS&E

PESCE, Mandya

# OVERVIEW

|  |
|--|
| Problem statement                              |
| Objectives                                     |
| System architecture                            |
| Modules developed<br>and their functionalities |
| Tools and technologies used                    |
| Testing approach used                          |
| Challenges and solution                        |

# PROBLEM STATEMENT

## **Advanced Cyberbullying Detection using ModernBERT**

Cyberbullying has become a significant issue due to the rise of social media and digital communication platforms. Traditional cyberbullying detection systems often focus solely on text-based content, failing to account for multimedia elements such as images and emojis. This limitation reduces the accuracy and effectiveness of these systems in identifying and mitigating cyberbullying incidents. The need for an advanced system that integrates both textual and visual content analysis is critical for improving cyberbullying detection accuracy.

# OBJECTIVES

1. detect harmful or offensive language on online platforms (gaming and social media), making online spaces safer .
2. To create a reliable system that can automatically flag potential cyberbullying by analyzing not just text, but also emojis and images, making detection more efficient and thorough
3. To achieve a model accuracy of above 90%, and compare with existing systems.

# SYSTEM ARCHITECTURE

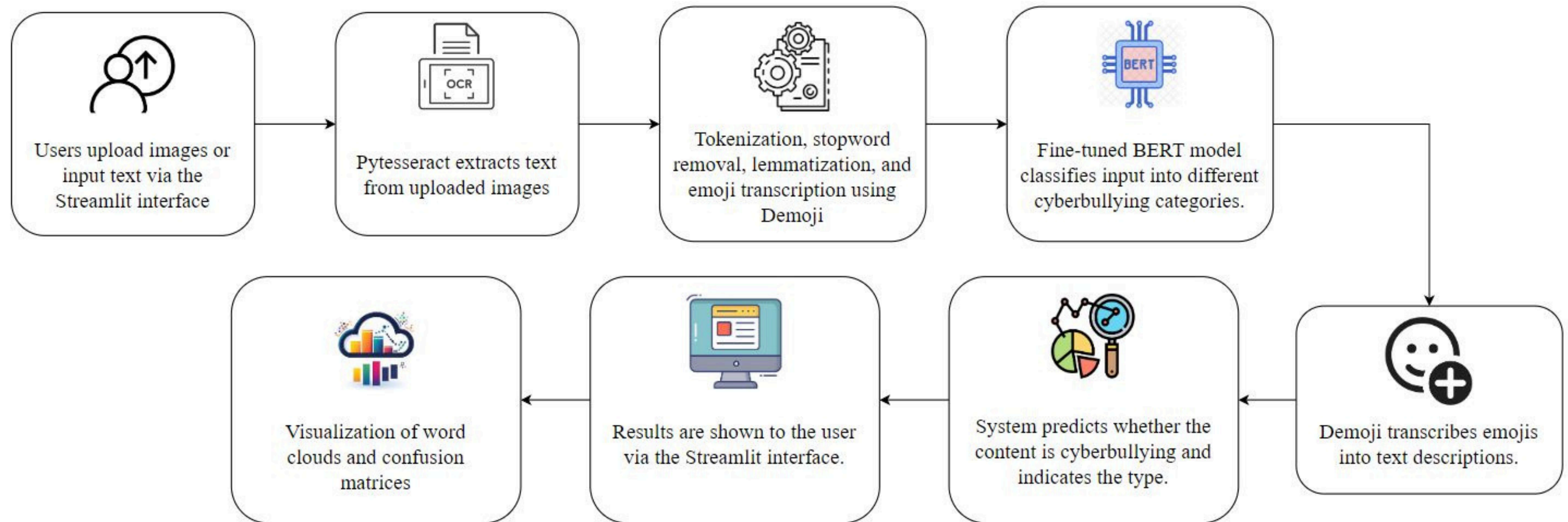
## High-Level System Design

1. **Data Preprocessing Module:** Extracts text from images using Pytesseract, removes noise, and processes emojis.
2. **Feature Extraction Module:** Uses demoji and NLP techniques to extract relevant features from text.
3. **Machine Learning Model:** Fine-tuned BERT model for classifying content as cyberbullying or non-cyberbullying.
4. **Prediction and Visualization Module:** Integrates with Streamlit to display results in an interactive interface.

## Low-Level System Design

1. **Text Processing Layer:** Tokenization, stemming, and stopword removal.
2. **Visual Content Analysis Layer:** Image OCR and emoji transcription.
3. **Deep Learning Model:** Fine-tuned BERT classifier.
4. **Result Interpretation and API Layer:** Displays predictions and logs results.

# SYSTEM ARCHITECTURE





# MODULES DEVELOPED AND THEIR FUNCTIONALITIES

1. **Data Collection Module** – Captures text and image-based content for analysis.
2. **Text Processing Module** – Cleans, tokenizes, and prepares text for analysis.
3. **Image Analysis Module** – Extracts text from images using Pytesseract.
4. **Emoji Processing Module** – Converts emojis into text for sentiment analysis.
5. **Machine Learning Module** – Uses a fine-tuned BERT model for classification.

# TOOLS AND TECHNOLOGIES USED

| Component                   | Technology Used                   |
|-----------------------------|-----------------------------------|
| Programming Language        | Python                            |
| Machine Learning Frameworks | TensorFlow, PyTorch, Scikit-learn |
| NLP Tools                   | NLTK, Transformers (BERT)         |
| OCR and Image Processing    | Pytesseract, PIL                  |
| Web Interface               | Streamlit                         |
| Data Processing             | NumPy, Pandas                     |
| Visualization               | Matplotlib, Seaborn               |
| Network Traffic Analysis    | Scapy, Wireshark                  |



# CHALLENGES AND SOLUTION

1) Handling emojis in cyberbullying detection

Solution: Integrated emoji library for emoji-to-text conversion, enhancing sentiment analysis and overall detection accuracy.

2) low software resource

solution: used Azure cloud for computing power

3) Noise and irrelevant symbols in tweets

Solution: Used regular expressions and text cleaning techniques to remove unnecessary symbols, links, and mentions (like @user, #hashtags).

# Azure machine learning workspace with serverless compute

The screenshot displays the Azure AI Machine Learning Studio interface. The top navigation bar shows 'Azure AI | Machine Learning Studio' and 'Azure subscription 1 MLworkspace'. The breadcrumb path is 'Default Directory > MLworkspace > Notebooks'. The notebook editor has several tabs: 'quickstart.ipynb', '\*onlineSafety.ipynb' (active), 'cyberbullying\_tweets.', and 'onlineSafety.html'. The 'Compute' dropdown is set to 'Serverless Spark Compute - Available', and the language is 'PySpark (Python)'. The notebook status is 'Ready (Stop session)'. The code cell [15] contains the following PySpark code:

```
1 tweet_data['cyberbullying_type'].value_counts()  
2
```

The output for cell [15] is:

```
[15] ✓ 1 sec - Command executed in 324 ms by pushpamadev1012 on 10:56:38 PM, 3/21/25  
  
religion          7998  
age               7992  
gender            7973  
ethnicity         7961  
not_cyberbullying 7945  
other_cyberbullying 7823  
Name: cyberbullying_type, dtype: int64
```

Below the output, there are tabs for '+ Code' and '+ Markdown'. The code cell [17] contains the following PySpark code:

```
1 def extract_emojis(text):  
2     emojis = re.findall(r'[\U0001F600-\U0001F64F\U0001F300-\U0001F5FF\U0001F680-\U0001F6FF\U0001F1E0-\U0001F1FF]', t  
3     return emojis  
4  
5 tweet_data['emojis'] = tweet_data['tweet_text'].apply(extract_emojis)  
6
```

The output for cell [17] is:

```
[17] ✓ 1 sec - Command executed in 886 ms by pushpamadev1012 on 10:57:38 PM, 3/21/25
```

# RESEARCH PAPER LINK

[https://docs.google.com/document/d/1QQW5zdNZ2Eaue8r2M2dU\\_bN0uqr8xldNoG0QpgxWtw4/edit?usp=sharing](https://docs.google.com/document/d/1QQW5zdNZ2Eaue8r2M2dU_bN0uqr8xldNoG0QpgxWtw4/edit?usp=sharing)



**THANK YOU**

