Hateful Memes Classification

Ahmad Abdul Muttal (22L-7466)
Affan Malik (22L-7533)

May 13, 2025

Introduction

With the increasing growth in social media users, the content variety diversifies with each user having a different perspective of the world, causing a major issue in the young vulnerable minds. Every post on any social media platform has an impact on one's thinking, whether it be topics like the Palestine War, the US elections, or even a celebrity's lifestyle. Memes—amusing videos or images—might feel like mere entertainment, but they are more than just a source of laughter. Hateful Memes Classification looks for such memes that might spread hate toward a specific group of people.

Basic Idea

Images are collected from various sources (currently Facebook), and the text is extracted and stored in jsonl format. The pre-existing jsonl files contain: id (image identifier), text (text in the image), image format, and label (0 for not hateful, 1 for hateful). Images are processed along with text being tokenized and passed through models to predict the label.

Data Visualization

Below are a few visualizations used to identify general patterns in the data for choosing the best model for prediction.

Visual Comparisons

Class Distribution

Non-hateful memes are almost twice as many as hateful ones as shown the figure 1, causing a minority problem in the classification of memes.

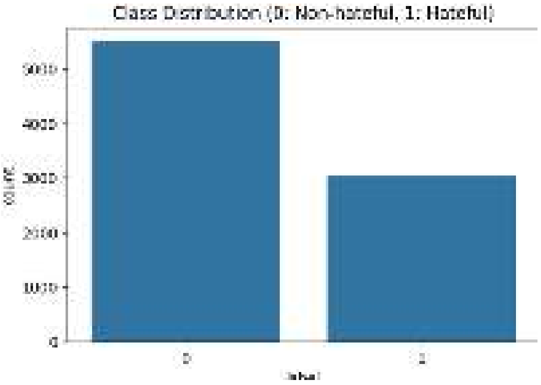


Figure 1: Class distribution of hateful vs non-hateful memes

Image and Text Representations

Distribution of text length:

Figure 2 shows that the text length is right-skewed because not a lot of text can be fit in a single frame.

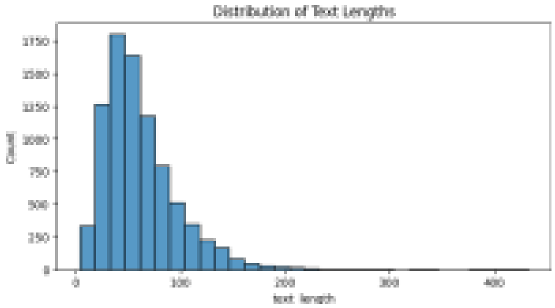


Figure 2: Distribution of text lengths in memes

WordCloud:

The WordCloud in figure 3 shows a lot of racial words along with slurs, making the understanding of sarcasm or context a challenge.

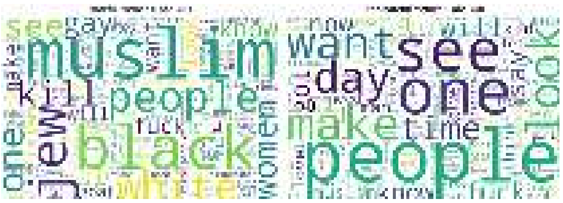


Figure 3: WordCloud of text extracted from memes

Distribution of Text Length vs Label:

A box and whiskers plot in figure 4 shows the relationship between text length and class label, suggesting that hateful memes have shorter text length.

ResNet50 + BERT Model Evaluation

Our first multimodal baseline combines a ResNet-50 image encoder with a pretrained BERT text encoder. We extract a 2048-dimensional feature vector from the penultimate layer of ResNet-50 and a 768-dimensional pooled output from BERT, concatenate them, and feed the result through a two-layer MLP classifier.

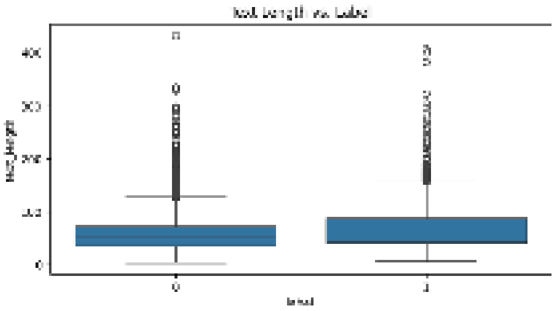


Figure 4: Boxplot of text length vs meme label

Evaluation Results

Metric	**Score**
Accuracy	0.605
ROC-AUC	0.605
Precision (macro avg)	0.57
Recall (macro avg)	0.56
F1-Score (macro avg)	0.56
Precision (weighted avg)	0.59
Recall (weighted avg)	0.60
F1-Score (weighted avg)	0.60

Table 1: Model Evaluation Results

Interpretation: An accuracy of 60.5% indicates that the model correctly classifies roughly three out of five memes, outperforming a random-guess baseline (50%). ROC-AUC of 0.605 shows modest separability. Macro-averaged F1 (0.56) reflects balanced performance, while weighted F1 (0.60) suggests better performance on the majority class.

EfficientNet + DistilBERT Model Evaluation

Our second multimodal baseline pairs an EfficientNet-B0 image encoder with a pretrained DistilBERT text encoder. A 1280-dimensional vector from EfficientNet and a 768-dimensional vector from DistilBERT are concatenated and passed through the same two-layer MLP.

Evaluation Results

Metric	Score
Accuracy	0.6235
ROC-AUC	0.6246
Precision (macro avg)	0.59
Recall (macro avg)	0.58
F1-Score (macro avg)	0.58
Precision (weighted avg)	0.61
Recall (weighted avg)	0.62
F1-Score (weighted avg)	0.62

Table 2: Model Evaluation Results

Interpretation: Accuracy improves to 62.35%. ROC-AUC indicates better separability. EfficientNet-B0 and DistilBERT reduce model size and complexity while improving performance, making this combination efficient and strong.

CLIP Model Evaluation

The CLIP baseline uses OpenAI's ViT-B/32 model to jointly encode image and text into a shared embedding space. Classification is done via cosine similarity between embeddings.

Interpretation: CLIP achieves 69.63% accuracy and 0.7068 ROC-AUC, outperforming CNN+Transformer baselines. Macro and weighted F1 scores indicate balanced performance. Pretraining allows strong results even with minimal fine-tuning.

Metric	Score
Accuracy	0.6963
ROC-AUC	0.7068
Precision (macro avg)	0.62
Recall (macro avg)	0.61
F1-Score (macro avg)	0.61
Precision (weighted avg)	0.63
Recall (weighted avg)	0.64
F1-Score (weighted avg)	0.63

Table 3: Model Evaluation Results

Conclusion & Model Comparison

Overall Performance:

- CLIP achieves the highest accuracy (69.63%) and ROC-AUC (0.707).
- EfficientNet-B0 + DistilBERT comes second (62.35% / 0.625).
- ResNet50 + BERT performs lowest (60.5% / 0.605).

Resource Efficiency:

- ResNet50 + BERT is the heaviest in resources.
- EfficientNet + DistilBERT offers a good trade-off.
- CLIP is large but benefits from extensive pretraining.

Ease of Deployment:

- CNN+Transformer models require fusion layers and tuning.
- CLIP can be used with minimal fine-tuning.

In summary, CLIP offers top-tier out-of-the-box performance, while EfficientNet-B0 + DistilBERT is ideal for balanced performance and resource efficiency.

Model Conclusions

1. ResNet50 + BERT Model Conclusion

The ResNet50 + BERT model has shown solid performance with an accuracy of 60.5% and a ROC-AUC of 0.605. These results are strong for a baseline model, showcasing the potential of combining ResNet50 for image features and BERT for text. This model demonstrates that multimodal learning can effectively integrate image and text data to solve complex tasks. The precision, recall, and F1 scores reveal that the model is performing well on the majority class while also making strides in improving the handling of the minority class. Given its performance, this model sets a strong foundation for future advancements in multimodal tasks, with improvements in fine-tuning and model integration continuing to boost its effectiveness.

2. EfficientNet + DistilBERT Model Conclusion

The EfficientNet + DistilBERT model marks a significant step forward, achieving an accuracy of 62.35% and a ROC-AUC of 0.6246. This model effectively balances efficiency with performance, demonstrating that lighter architectures like EfficientNet-B0 and DistilBERT can yield strong results even with fewer parameters. The increase in accuracy over the ResNet50 + BERT model showcases how more optimized architectures can maintain, and even improve, performance while reducing computational load. The precision, recall, and F1 scores reflect the model's continued progress in managing class balance effectively. This model proves the value of efficient design in achieving improved results without compromising on the task's complexity.

3. CLIP Model Conclusion

The CLIP model, utilizing the pre-trained ViT-B/32 architecture, delivers exceptional performance with an accuracy of 69.63% and a ROC-AUC of 0.707. Given the nature of the dataset, this performance is highly impressive and speaks to the strength of CLIP's pretraining on large-scale multimodal data. CLIP excels by integrating image and text features in a shared space, outperforming previous models in both accuracy and robustness. The model's ability to perform exceptionally well with minimal fine-tuning is a testament to the power of large-scale pretraining and its generalizability across diverse tasks. With its high accuracy, CLIP stands as a powerful tool for multimodal tasks, showing substantial promise in the context of hate speech detection and other complex problems.

Comparison & Visualization

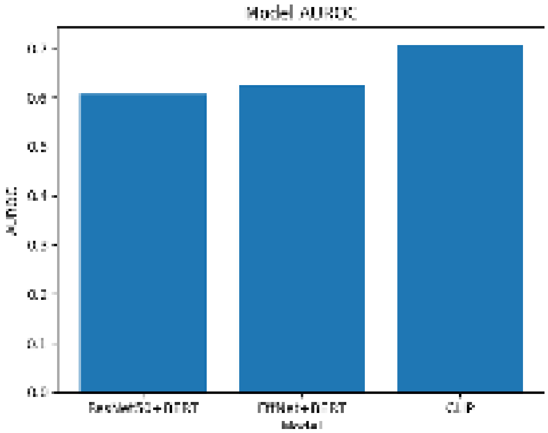


Figure 5: Model AUROC

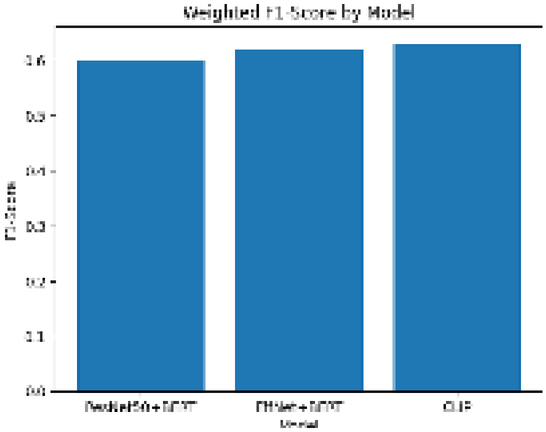


Figure 7: Weighted F1-Score by Model

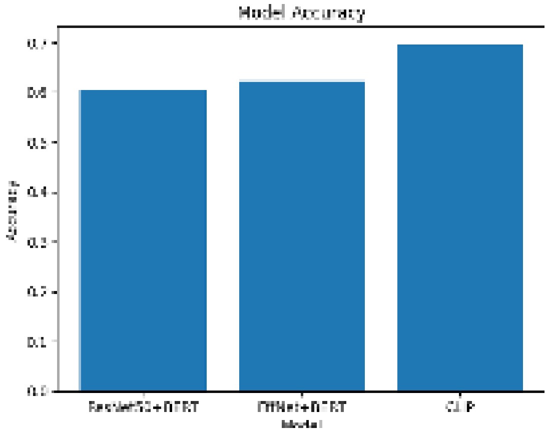


Figure 6: Model Accuracy

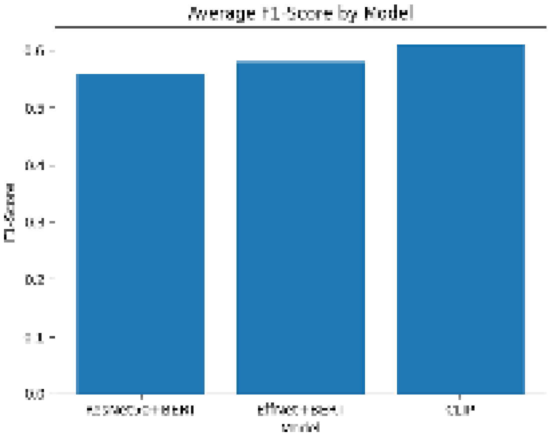


Figure 8: Average F1-Score by Model

Explainable AI Methods

1. Integrated Gradients

Definition: Attributes each input feature's contribution by integrating the gradients along the straight-line path from a baseline (e.g., an all-zero image) to the actual input.

Strengths:

- Axiomatic guarantees (completeness, sensitivity): attributions sum to the model's output difference from baseline.
- Baseline flexibility: choose references suitable to your domain (e.g., blurred image, black image).

Use Cases:

- Pixel-level attribution in image classifiers.
- Feature importance in tabular models (e.g., which demographic factors drive a credit-risk score).

Advantages:

- Produces smooth, low-noise attributions.
- More robust to gradient saturation than plain gradients.

2. Layer-GradCAM

Definition: Generates class-specific heatmaps by weighting the activation maps of a chosen convolutional layer by the gradients of the target class with respect to those activations.

Strengths:

- Spatial localization: highlights which regions the network "looked at."
- Layer choice: early layers reveal fine details; later layers capture high-level concepts.

Use Cases:

- Identifying object parts driving classification decisions (e.g., wheels vs. body of a car).
- Debugging misclassifications by checking focus regions.

Advantages:

- Class-discriminative: focuses on features relevant to the predicted class.
- Easy to overlay on original images for intuitive visualization.

3. Saliency Maps

Definition: Computes the gradient of the output score with respect to each input feature, highlighting where small input changes have the biggest effect.

Strengths:

- Simplicity: one of the earliest gradient-based methods; very fast to compute.
- General applicability: works for any differentiable model.

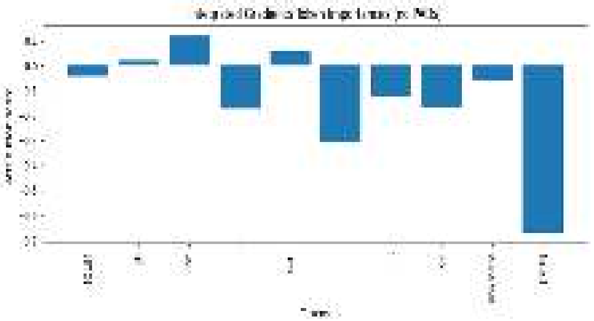
Use Cases:

- Quick sanity checks to verify the model uses reasonable pixels or features.
- Baseline for comparison with more sophisticated methods like Integrated Gradients.

Advantages:

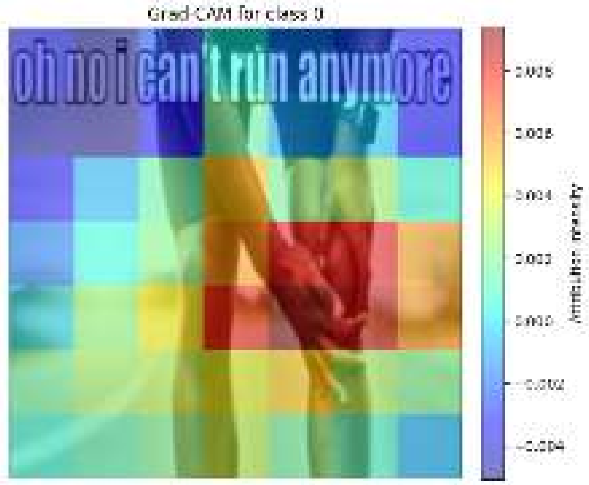
- Computationally lightweight—ideal for rapid prototyping.
- Provides a first-glance sense of input sensitivity.

Inferences(ResNET50+BERT):



Inference:

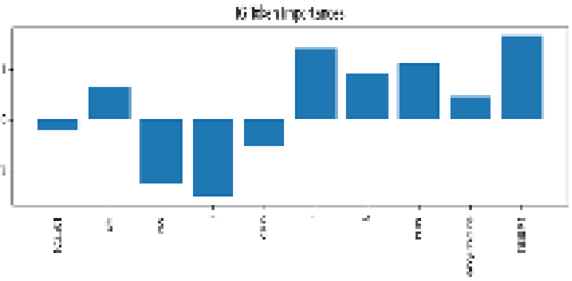
- “no” has the highest positive attribution (~+0.12), so its presence pushes the prediction toward the chosen class most strongly.
- Higher the value, more its contribution towards its predicted class



Inference:

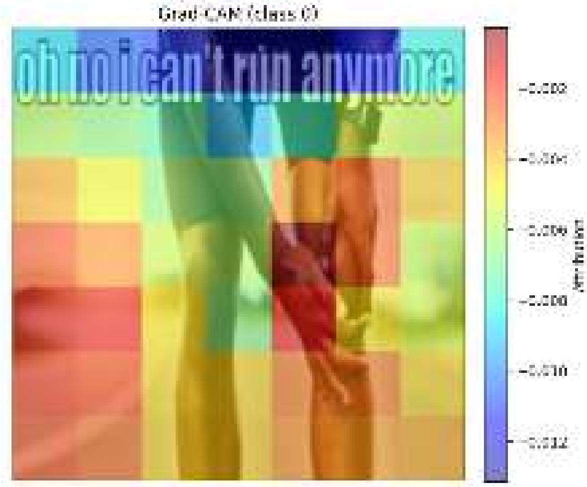
- The image generated by Layer GradCAM shows the knee area the most significant in predicting its class as non-hateful(0)
- Blue area shows less impactful towards its prediction

Inferences(EfficientNET + DistilBERT):



Inference:

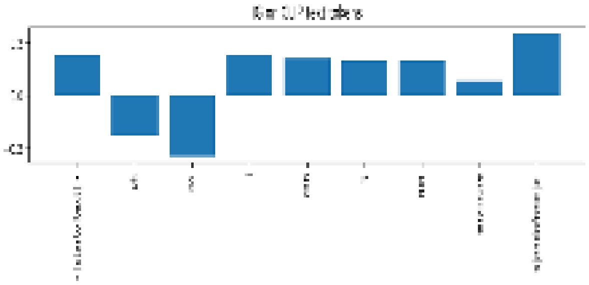
- “run” has the highest positive attribution (~+0.10), so its presence pushes the prediction toward the chosen class most strongly.
- Words “no” and “i” contribute less towards the predicted class



Inference:

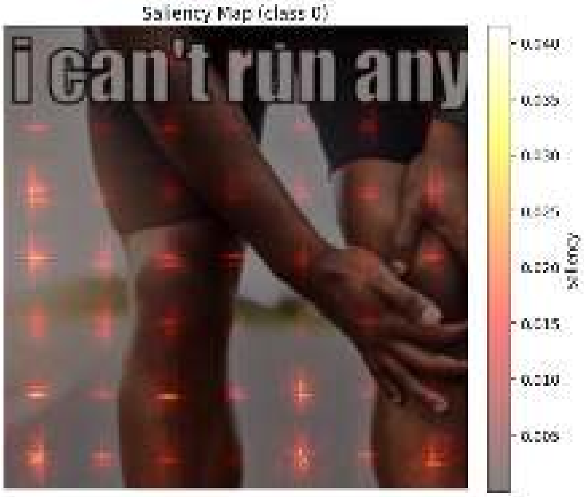
- The red spots in the image indicates the area of image which is more significant in predicting its class 0
- Blue and Green area shows less impactful towards its prediction

Inferences(CLIP):



Inference:

- “i”, “can”, “run” has the highest positive attribution among all(excluding <|endoftext|>) so its presence pushes the prediction toward the chosen class most strongly.
- Words “no” and “oh” contribute less towards the predicted class



Inference:

- The bright spots in the image shows the most attributing area towards its predicted class 0.
- The text part is overlooked, hence result in less noise.