Identifying Hateful Memes

Sandip Subedi ssubedi33@gatech.edu
Gagan Dangi gdangi3@gatech.edu
Abdul Moiz Amir aamir37@gatech.edu
Zhuoxun Yang zyang668@gatech.edu

Abstract

Hateful memes pose a significant challenge in content moderation due to their multimodal nature. The sarcasm, cultural context, or implicit meaning in memes frequently arise from the interplay between the image and the caption. This project explores the classification of hateful memes using three different approaches: an image-based classifier, a text-based classifier, and a multimodal CLIP-based classifier that integrates both image and text features through different fusion strategies—concatenation, ensemble, and self-attention—followed by a deep neural network classifier. The primary objective is to maximize the detection of hateful memes while minimizing false positives on non-hateful content.

1. Introduction

The rapid rise and spread of online content in this era has created an urgency to develop automatic identification of the hateful memes — the combination of the image and text that makes the visual content spreading harmful stereotypes, discrimination, or incite violence. These memes when viewed textually or visually in isolation often look benign and this creates the challenge to classify the memes as hateful or not. The aim in this study is to explore a variety of architectures (unimodal and multimodal) and compare among themselves and baseline to discover the approach which best captures the joint semantics for classification task.

It is common that the hate detection is often approached via Natural Language Processing models applied to the textual cues [2]. Nonetheless, such models often fail to identify the hateful intent when either image or text is not overly harmful in isolation [6]. The present image classifiers are good at object or scene recognition, however they are inefficient in interpreting the symbolic hate like gestures. The multimodal approaches are dependent on the naive concatenation of image and text embeddings or taking into account each modality in isolation which becomes the limiting factor for modeling the cross-modal dependencies [13]. Further, the class imbalance in the dataset also becomes a challenge in ensuring efficient performance for the classfication task [6]. There still remains some gap in designing robust attention-guided and concept-aware fusion strategies curated particularly for hate detection.

The solution to hate detection problem is pivotal in proactively moderating the content on social media platforms. The robust solution can be useful in automatic flagging of the nuanced multimodal hate speech, easing manual moderation burden, and eventually protecting users from abusive and hateful content. In broader sense, it can contribute to the adjacent tasks such as misninformation detection, sarcasm detection, media forensics, and cross-cultural content understanding.

This work utilizes the Hateful Memes Challenge Dataset [6] that has 10000 human-annotated memes with both visual and textual components. The challenge in the dataset is that numerous examples can pass as benign in one modality and can be hateful or harmful when both modality are paired. The imbalance in data due to only 35% being hateful is another challenge for traditional classifiers and this calls for both precision and recall. This dataset also enables the rigorous evaluation of both unimodal and multimodal models in identifying the context-dependent hate.

2. Approach

Our overall approach to hateful meme classification is structured into two main sections: unimodal and multimodal. The unimodal baselines are to investigate efficacy of text or image features individually in classifying hateful memes. The goal of our multimodal approach is not only to improve classification performance with the integration of image and text fearures, but also to investigate the efficacy of different fusion mechanisms in capturing cross-modal interactions essential for hateful meme understanding.

2.1. Unimodal

We construct two unimodal classifiers: one based solely on text and another based solely on image.

2.1.1 HateBERT - Text Based Classifier

For the baseline depending on text alone, we fine-tune the HateBERT model. HateBERT is a RoBERTa based model pre-trained on 57 million hateful Reddit comments [1]. Captions are lower-cased and passed through HateBERT's own tokenizer. Given the class-imbalance, the class weights are determined via inverse frequency, $w_k = \frac{N}{C\, n_k}$ (where $C = 2$), and are injected into a weighted cross-entropy loss:

$$\mathcal{L}_{\text{WCE}} = -\frac{1}{B}\sum_{i=1}^{B} w * y_i \log p_{i,y_i}, \tag{1}$$

with minibatch size B, true label y_i, and model probability $p_{i,y_i} = P_\theta(y = y_i \mid x_i)$

2.1.2 CA_CNN - Image Based Classifier

We propose a confidence-aware CNN architecture that performs joint optimization of classification accuracy and uncertainty estimation. Our approach introduces a specialized confidence head comprising progressively smaller layers (256→64→1 neurons with batch normalization) designed to prevent overfitting while maintaining sufficient capacity for uncertainty estimation. Unlike Monte Carlo Dropout [4] and Deep Ensembles [7], which require multiple forward passes during inference, our confidence head provides uncertainty estimates in a single forward pass, making it computationally efficient for real-time content moderation deployment.

The network is trained using a composite loss function:

$$L_{\text{total}} = L_{\text{classification}} + \lambda \cdot L_{\text{confidence}} \tag{2}$$

where $\lambda = 0.1$ balances classification accuracy and uncertainty quality while trading theoretical rigor for practical feasibility.

We implement systematic baseline exploration using predetermined configurations: conservative ($\lambda = 0.1$, lr $= 1 \times 10^{-4}$), aggressive ($\lambda = 0.3$, lr $= 3 \times 10^{-4}$), and stable ($\lambda = 0.1$, lr $= 5 \times 10^{-5}$). Analysis on 1,000 test samples revealed that ResNet-50 with conservative configuration achieved optimal performance (AUC-ROC: 0.5714, representing a 3.9% statistically significant improvement, $p < 0.05$), with a confidence-accuracy correlation of 0.051, demonstrating 1.9× better uncertainty quantification than EfficientNet-B0 (0.045 vs 0.027). Critically, increasing the confidence loss weight λ beyond 0.1 degraded uncertainty estimation quality, revealing optimization conflicts inherent in joint training objectives.

High-confidence threshold analysis demonstrated practical selective prediction potential, achieving 58.1% accuracy on 48.7% of cases. However, an Expected Calibration Error (ECE) of 0.169 indicates overconfidence bias requiring further calibration. This framework enables deployment strategies where 85% of content could be automated while flagging the remaining 15% for human review, providing a principled foundation for understanding when image-only approaches require multimodal augmentation.

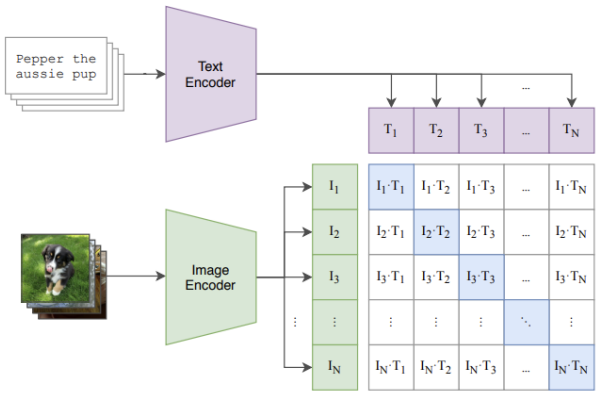

Figure 1: Diagram that shows CLIP training Approach [10]

2.2. Multimodal

In our multimodal approach, we use both the image and the caption present in memes to create our classifier. A pretrained Contrastive Language-Image Pretraining (CLIP) model is used as the feature extraction backbone. The extracted features are integrated using multiple fusion strategies, including concatenation, ensemble, and self-attention. These fused representations are then used to train a deep neural network classifier for meme classification.

2.2.1 CLIP

CLIP is a multimodal neural network developed by OpenAI that connects image and text. It is trained on 400 million image-text pairs collected from a variety of publicly available sources. CLIP learns a multimodal embedding space by jointly training an image encoder and text encoder to maximize the cosine similarity of image and text embeddings of correct pairs while minimizing the cosine similarity of incorrect pairs [10]. Figure 1 illustrates how CLIP jointly trains an image encoder and text encoder. $I_1, I_2, I_3, \ldots, I_N$ denote the embeddings for images, and $T_1, T_2, T_3, \ldots, T_N$ represent the corresponding text embeddings. The correct image-text pairs denoted as $I_1T_1, I_2T_2, I_3T_3, \ldots, I_NT_N$ are aligned along the diagonal in Figure 1. During training, the model maximizes the cosine similarity for these correct pairs while minimizing the the cosine similarity for all other incorrect pairs.

For our hateful meme classifier, a pre-trained CLIP model (ViT-L/14 to encode image and Transformer-based encoder

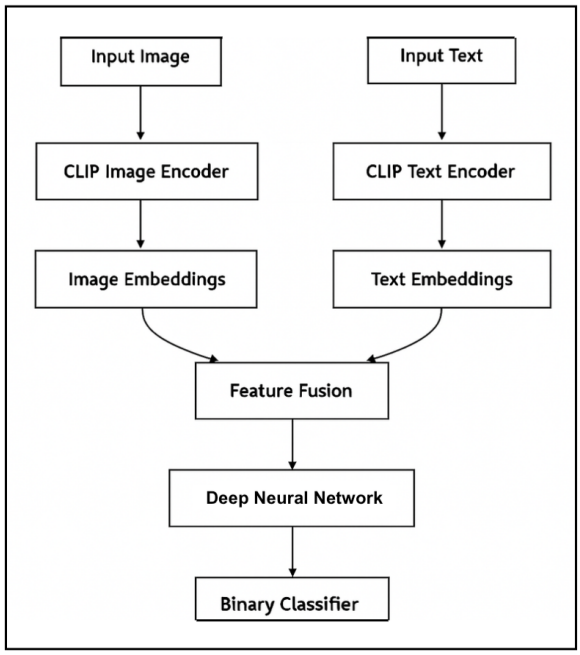

Figure 2: CLIP Based Hateful Meme Classifier

for text) is used to extract image embeddings from the image and text embeddings from the caption. Figure 2 shows the model pipeline. The image and text embeddings are fused together using concatenation and ensemble. We also employ attention based extension to the CLIP with an addition of lightweight self-attention head. A deep neural network is trained with the features obtained from these fusion strategies for meme classification. The models with concatenation, ensemble, and self-attention are named ConcatCLIP, EnsembleCLIP, and AttentionCLIP respectively.

2.2.2 ConcatCLIP

Both image and text embeddings obtained from CLIP are simply stacked together into a column vector, creating a single feature representation vector from both image and text. The deep neural network classifier is designed to map the feature representation vector to a binary prediction score. It consists of three fully connected layers with intermediate dimensions expanding from d to $2d$, then compressing back to d, and finally projecting to a scalar output. d is the size of the vector representing the features. Each linear layer (except the last) is followed by the Gaussian Error Linear Unit (GELU) activation function, which introduces smooth non-linearities. We apply dropout after the first two GELU activations to prevent overfitting. Layer normalization is used after each dropout to stabilize the training by normalizing activations across the feature dimension. This approach is motivated by its simplicity while enabling the classifier to learn cross-modal interactions.
We use weighted Binary Cross Entropy (BCE) Loss shown in equation (3) to train ConcatCLIP.

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[w * y_i \log \sigma(\hat{y}_i) + (1 - y_i)\log(1 - \sigma(\hat{y}_i))] \tag{3}$$

where:

- N is the batch size
- w is the class weight for imbalance correction
- $y_i \in \{0, 1\}$ is the true label (1=hateful)
- $\hat{y}_i$ is the model's logit output
- $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function

The weighting increases the loss contribution from hateful samples (minority class) which prioritizes model to learn minority class.

2.2.3 EnsembleCLIP

Two separate classifiers (one with CLIP image embeddings and another with CLIP text embeddings) are designed independently and then combined via learnable weight α ($\alpha \cdot P_{\text{img}} + (1 - \alpha) \cdot P_{\text{text}}$). Each classifier is composed of three fully connected layers that expand the input embedding from d to $2d$, compress it back to d, and finally project it onto a scalar output. Here, d represents the dimension of the embeddings. Each linear layer (except the final one) is followed by normalization and the Gaussian Error Linear Unit (GELU) activation function. GELU introduces smooth non-linearities which allows model to learn complex patterns. We apply dropout after the first two GELU activations to prevent overfitting. We use Batch normalization for image embedding and Layer Normalization for text embedding to stabilize the training. Ensembling allows the model to adaptively balance the contribution of each individual models and leverage complementary information. We optimize EnsembleCLIP using the same loss function from equation (3).

2.2.4 AttentionCLIP

The AttentionCLIP version is the extension of the standard CLIP pipeline featuring an addition of lightweight self-attention head to re-weight the projected 25 category-level cosine similarities before the fusion with the 768-D image embedding. Post the L2-normalization of both image and text vectors, the cosine-similarities against 225 hate prompts are calculated. During the process, each category's

nine prompts are collapsed to a single max-score, resulting in 25 values which are projected to 128-D vector. The categorical representations are re-weighted by a Multi-head self-attention layer which enables the model to have dynamic focus on the most relevant hate concepts per meme. The attended 128-D vector and the 768-D image embedding are concatenated and passed through a 1024-512-1 MLP to yield the final probability. The key novelty in this approach is the application of self-attention directly to category-wise CLIP similarities instead of to raw pixels or tokens. The final prediction is optimized using Focal Loss in equation (4) , addressing the class imbalance, for dynamic down-weight of the easy negative samples and emphasize on training the tough samples.

$$\mathcal{L}_{\text{focal}} = \frac{1}{N} \sum_{i=1}^{N} \alpha \cdot (1 - p_t^{(i)})^{\gamma} \cdot \text{BCE}(x^{(i)}, y^{(i)}) \tag{4}$$

where:

- N is the batch size
- $x^{(i)}$ is the logit output for sample i
- $y^{(i)} \in \{0, 1\}$ is the ground-truth label
- $p^{(i)} = \sigma(x^{(i)}) = \frac{1}{1+e^{-x^{(i)}}}$ is the sigmoid probability
- $p_t^{(i)} = \begin{cases} p^{(i)} & \text{if } y^{(i)} = 1 \\ 1 - p^{(i)} & \text{if } y^{(i)} = 0 \end{cases}$
- $\alpha \in [0, 1]$ is a class-balancing weight
- $\gamma \geq 0$ is the focusing parameter
- $\text{BCE}(x^{(i)}, y^{(i)}) = -y^{(i)} \log p^{(i)} - (1 - y^{(i)}) \log(1 - p^{(i)})$

2.2.5 Evaluation Metrics

We evaluate model performance using five standard metrics:

- **Accuracy**: It is the ratio of total correct prediction to total number of samples.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

- **AUC-ROC**: It is the area under the Receiver Operating Characteristic curve (range: 0.5–1.0) which measures class separability across all thresholds.
- **Precision**: It is the ratio of correctly predicted positive instances among all instances predicted as positive.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

- **Recall**: It is the ratio of correctly predicted positive instances among all actual positive instances.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

- **F1-score**: It is the harmonic mean of precision and recall.

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives respectively. For hateful meme classification, we prioritize high recall (minimizing false negatives) while maintaining reasonable precision, with F1-score providing a balanced measure. We optimize the F1-score on the validation set.

3. Experiments

We conduct our experiments using the dataset from Phase I of the Facebook Hateful Memes Challenge. The dataset is divided into three parts: a training set containing 8,500 memes (35% labeled as hateful and 65% as non-hateful), a validation set with 500 memes (50% hateful and 50% non-hateful), and a test set of 1,000 memes (50% hateful and 50% non-hateful).

3.1. HateBERT

Using Optuna (20 trials, sequential), we explore:

- *learning-rate* $\in [1\times10^{-5}, 7\times10^{-5}]$ (log-uniform)
- *weight-decay* $\in [0, 0.3]$
- *warm-up steps* $\in [0, 500]$
- *epochs* $\in \{2, 3, 4\}$
- *batch size* $\in \{8, 16\}$

This space was determined following prior work on BERT fine-tuning, which has shown to converge in three passes for small data sets [3, 8, 11]. The objective is macro-F1 on the validation split to determine the best parameters. However, for testing, the binary average is used. The winning setting is: LR $= 4.9 \times 10^{-5}$, WD $= 0.15$, warm-up $= 157$, epochs $= 2$, batch $= 16$. Lastly, the threshold is calibrated to 0.3 during testing to maximize F1.

3.2. CA_CNN

This section initiates a comparison of the ResNet-50 [5] and EfficientNet-B0 [12] architectures both enhanced with confidence estimation. The evaluation focused on three dimensions: classification performance quality, uncertainty quantification capability, and coverage-accuracy trade-offs.

The evaluation included standard metrics such as accuracy, AUC-ROC, and F1-score, along with Expected Calibration Error (ECE) [9] and confidence-accuracy correlation coefficients. All experiments were conducted using PyTorch 2.6.0+cu124 and implemented bootstrap resampling with 1,000 iterations for statistical validation. The confidence head adds negligible computational overhead (<5% training time increase) while providing deployment-critical uncertainty information.
We hypothesized that architectural efficiency would systematically influence the capacity for uncertainty quantification, leading us to conduct a systematic exploration of parameters focused on the dynamics of confidence-weight optimization.

3.3. ConcatCLIP

The hyperparameters were tuned to reduce the BCE loss on the training set. The threshold was tuned to optimize the F1-score on the validation set. The optimized setting is: learning rate = 0.0001, weight = 1.86, epochs = 10, batch size = 32, threshold = 0.01. The model converged with a training loss of 0.0885 with exceptional AUC-ROC of 0.9967. On our validation set of 500 memes, it maintained robust performance with AUC-ROC of 0.8088, demonstrating good generalization despite the imbalanced training set and small validation size.

3.4. EnsembleCLIP

The hyperparameters were tuned to reduce the BCE loss on the training set. The threshold was tuned to optimize the F1-score on the validation set. The optimized setting is: learning rate for model weights = 0.0001, learning rate for α = 0.1, weight = 1.86, epochs = 15, batch size = 32, threshold = 0.1. The model converged with a training loss of 0.0787 with exceptional AUC-ROC of 0.9973. On our validation set of 500 memes, it maintained robust performance with AUC-ROC of 0.7390. Although the training loss and AUC-ROC for EnsembleCLIP were better than ConcatCLIP, it performed worse on the validation set which suggests that EnsembleCLIP is prone to overfitting.

3.5. AttentionCLIP

The approach and architecture for this model has been explained in detail in section 2.2.4. The hyperparameter optimization was done using Optuna over 12 trials with 10 epochs in each trail and the objective was chosen to be the validation AUC. The search ranges were: $[1 \times 10^{-5}, 1 \times 10^{-3}]$ for the learning rate (LR), $[1 \times 10^{-5}, 1 \times 10^{-2}]$ for weight decay, and $[0.25, 0.75]$ and $[1.0, 3.0]$ for the focal loss parameters α and γ respectively. The first choice of loss was Binary Cross-Entropy Loss(BCE) which underperformed due to class imbalance in the data used. To counter this, the model was transitioned to Focal Loss such that the model emphasizes learning the hard examples and down-weighs the easy negatives. The training mechanism also featured the early stopping on validation AUC with patience of 6. The best model was selected at epoch 24, reaching the AUC of 0.7356. Further, the optimal threshold for binary classification was calibrated to maximize the F1-score on the validation set, achieving the optimal threshold of 0.343. The initial concerns were over-fitting on 8500 images and heavy class imbalance. With concatenation of all 225-cosine scores, there was over-fitting and plateau on validation AUC at around 0.68. With the implementation of the projected-attention in combination with focal loss and tuned dropout, the training stabilized and the validation AUC reached to around 0.7356.
To better understand the impact of each categories, prompt ablation study was also performed and the metrics were re-computed. The result is shown in Figure 5. Moreover, the error analysis was done to generate False positives and False Negative to analyze the efficiency and performance of the model.

4. Results

Table 1 presents the performance metrics of our models on the test set of 1000 memes. We evaluate each model using Accuracy, AUC-ROC, Precision, Recall, and F1 Score.

HateBERT The text-only model achieves moderate performance with an accuracy of 62.1% and an F1 score of 0.639. This result indicates that the textual content in hateful memes provides relatively strong cues for classification, especially since hateful intent is often more explicitly conveyed in text.

CA_CNN The image-only model performs the worst across all metrics, achieving an accuracy of just 51.7% and an F1 score of 0.535. This suggests that visual content alone is insufficient to detect hateful intent in memes, possibly due to the subtlety or ambiguity of visual signals without corresponding textual context.

ConcatCLIP ConcatCLIP, which combines image and text embeddings via feature concatenation, performs the best overall, with the highest accuracy (72.6%), AUC-ROC (0.8088), and F1 score (0.7410). The strong performance reflects the complementary nature of text and visual modalities when jointly modeled, allowing the network to learn a richer representation.

EnsembleCLIP The ensemble-based fusion method also performs well, yielding balanced precision and recall (both at 0.70). Although its F1 score (0.70) is slightly lower than

Model	Accuracy	AUC-ROC	Precision	Recall	F1 score
HateBERT	0.621	0.675	0.599	0.686	0.639
CA_CNN	0.517	0.531	0.506	0.567	0.535
ConcatCLIP	**0.7260**	**0.8088**	0.6901	0.8000	**0.7410**
EnsembleCLIP	0.7070	0.7658	**0.70**	0.70	0.70
AttentionCLIP	0.646	0.737	0.589	**0.916**	0.717

Table 1: Performance comparison of hateful meme classifiers

ConcatCLIP, its stable and symmetric performance across metrics makes it a reliable alternative fusion approach.

AttentionCLIP AttentionCLIP achieves the highest recall (0.916), indicating it is very sensitive in detecting hateful content. However, this comes at the cost of lower precision (0.589), meaning it produces more false positives. The high recall may be desirable in contexts where minimizing the risk of missing hateful content is critical, even at the expense of occasional over-flagging. The false positives and false negatives examples are presented only on the notebook due to extreme hateful content.

5. Conclusion and Future Work

In this work, we addressed the challenging problem of hateful meme classification by experimenting with unimodal and multimodal approaches. Our results demonstrate that multimodal fusion models significantly outperform unimodal models which highlights the importance of jointly modeling both image and text features in the meme. Among all approaches, the ConcatCLIP model achieved the highest overall performance in terms of accuracy and F1 score. AttentionCLIP model achieved the highest recall which makes it particularly effective for sensitive applications where the cost of missing hateful content is high. As we had hypothesized initially, the poor performance of the unimodal models highlights that both image or text feature alone cannot individually yield good results. This project calls for future works on the exploration of the more advanced fusion techniques such as cross-modal attention, improved calibration for recall-precision balance, and better prompt engineering for better generalization. The possible future directions can be the integration of the robust image augmentation for improving the better capture of the visual context, expansion of the scope to multilingual and geography-based memes, and incorporation of the interpretability methods.

6. Work Division

Sandip Implemented ConcatCLIP and EnsembleCLIP, and trained them with weighted Binary Cross Entropy Loss.

Abdul Moiz Amir Implemented the HateBERT fine-tuning pipeline with class-imbalance weighting and Optuna hyper-parameter search.

Zhuoxun Implemened a confidence-aware CNN architecture that enhances traditional hateful meme detection by adding uncertainty quantification capabilities, eventually achieving 0.5714 AUC-ROC with ResNet50 while enabling selective prediction based on model confidence scores.

Gagan Implemented Prompt-based CLIP model with Self-attention to the similarity vectors with focal loss and Optuna hyper parameters search, ablation study, and error analysis.

References

[1] Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. HateBERT: Retraining BERT for abusive language detection in english. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25. Association for Computational Linguistics, 2021. 2

[2] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*, 2017. 1

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL–HLT 2019*, pages 4171–4186, 2019. 4

[4] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1050–1059, 2016. 2

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[6] Douwe Kiela, Hamed Firooz, Vedanuj Mohan, Amanpreet Goswami, Anurag Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. In *NeurIPS Workshop on Multimodal Learning and Applications*, 2020. 1

[7] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, volume 30, 2017. 2

[8] Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. On the stability of fine-tuning BERT: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations (ICLR 2021)*, 2021. 4

[9] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *AAAI*, volume 29, pages 2901–2907, 2015. 5

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. 2, 8

[11] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune BERT for text classification? In *Chinese Computational Linguistics (CCL 2019)*, pages 194–206. Springer, 2019. 4

[12] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114, 2019. 4

[13] Pei Zhou, Peng Qi, Shuohang Zheng, Jiaming Wu, and Lu Huang. Multimodal transformers for detecting multimodal hate speech. In *Findings of the Association for Computational Linguistics: EMNLP*, 2021. 1

Appendix

The appendix provides supplementary material that supports and extends the findings presented in the main paper. It includes additional experimental results, ablation studies, visualizations, and implementation details that offer deeper insights into the proposed methods.

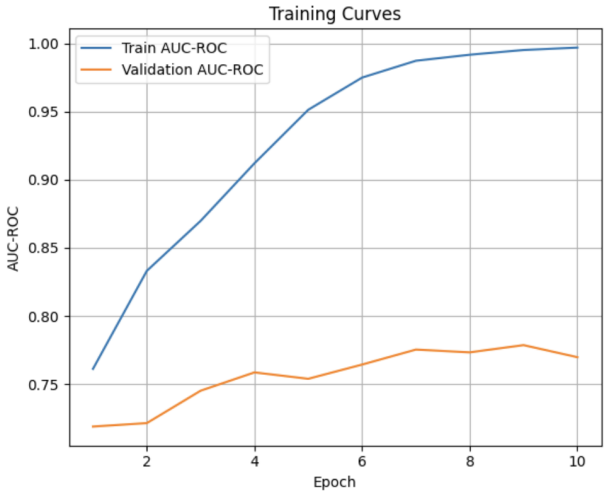

Figure 3: The evolution of AUC-ROC on train and validation while training our best model (ConcatCLIP). The model was allowed to learn everything from the training set.

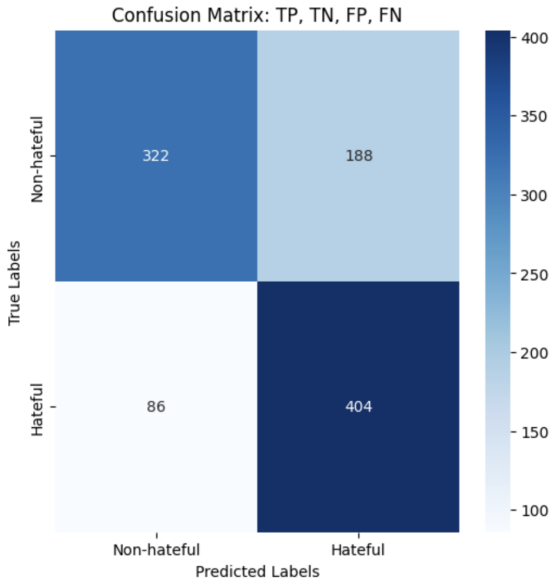

Figure 4: The confusion matrix on the test set from our best model (ConcatCLIP).

	AUC	Acc	F1	Dropped Indices
0	0.734308	0.662	0.713073	[0]
4	0.727641	0.637	0.706073	[20, 21]
5	0.726277	0.652	0.703578	[0, 1, 2, 3, 4]
2	0.718982	0.616	0.696682	[1, 2, 5]
3	0.718455	0.656	0.696649	[10, 11, 12]
1	0.718193	0.637	0.699254	[0, 3, 7]

Figure 5: Ablation study results on the Hateful Memes test set. When dropping indices 0, 3, and 7 (white-supremacist, antisemitic, and trans-exclusionary cues), the AUC drops the most, indicating their importance for accurate classification; modern-hate vectors show smaller effects.

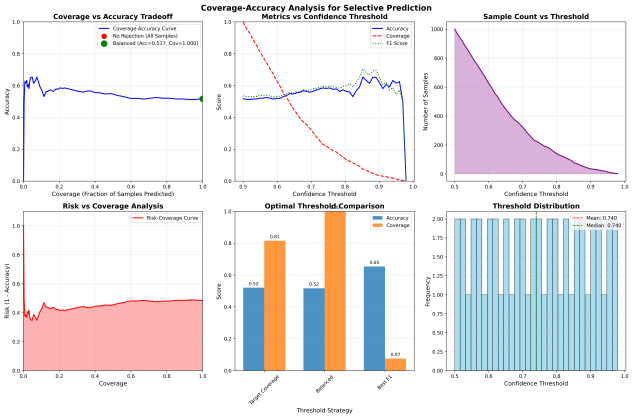

Figure 6: Coverage-accuracy analysis for CNN-based classifier showing confidence threshold effects on selective prediction performance [10].

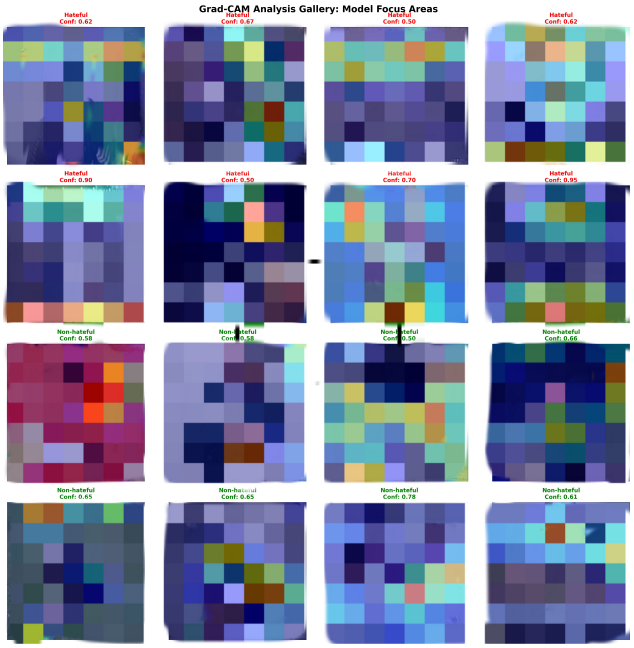

Figure 7: Grad-CAM attention visualizations for CNN-based hateful meme classification with confidence scores. Hateful memes are blurred.

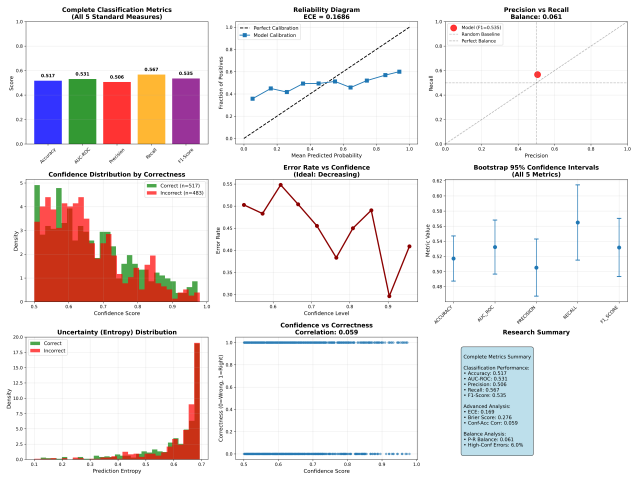

Figure 8: Comprehensive evaluation metrics and calibration analysis for CNN-based hateful meme classifier.