Final Project Report: Hateful Memes

Vedesh Yadlapalli
Georgia Institute of Technology
vedeshy@gatech.edu

Michael Osmolovskiy
Georgia Institute of Technology
mosmolovskiy@gatech.edu

Justin Xia
Georgia Institute of Technology
jxia74@gatech.edu

Soham Samal
Georgia Institute of Technology
ssamal31@gatech.edu

Abstract

Detecting hateful content within internet memes presents a unique challenge, as the harmful meaning often arises from the complex interplay between visual and textual elements, rather than from one modality alone. Accurate automated detection is essential to maintain safe online environments. This project explores multimodal deep learning approaches to classify memes from the Hateful Memes dataset. Our initial baseline methods struggled significantly (AUC of 0.58, F1 of 0.14). We experimented with transfer learning, different multimodal fusion techniques, data augmentation, weighted sampling, and auxiliary labeling strategies. Ultimately, we settled on a BLIP-based approach with a lexicon heuristic and data augmentation, achieving an F1-score of 0.65 and a recall of 0.89, correctly identifying the majority of hateful memes. These findings highlight BLIP's potential for this task, showing that combining multimodal features is important for understanding the entire meaning of a meme.

1. Introduction

Internet memes, which combine images and text, are often used to mix humor with cultural commentary. However, this medium can be used for the spread of hateful messages. The difficulty in making the distinction between comedy and hate is rather difficult for memes, as oftentimes the hateful meaning cannot be determined through the meme's image or text alone, but rather the combination of the two. Thus, our primary objective was to develop and train a neural network capable of analyzing multimodal inputs, in this case images and text, and accurately determine if it constitutes hate speech.

The ability to quickly, effectively, and accurately identify hateful content is extremely important in today's digital society. The ever-increasing growth of content production and propagation means that manual moderation is not enough to maintain safe online environments. Thus, automated systems are required to moderate the vast number online posts made each day. By finding a way to automatically detect and filter out harmful content, online platforms can protect their users from hate speech and foster a more positive online environment.

2. Related Works

Prior work has used pre-trained models like CLIP [13] to leverage large-scale vision-language representations for similar tasks. Additionally, research on multimodal fusion and attention-based architectures has shown promise in combining disparate data sources for improved classification accuracy [4, 16]. Kumar [7] was able to design an architecture using a CLIP multimodal architecture, achieving an AUROC score of 85.8 on the Hateful Memes Challenge dataset [6]. By extracting the image and the text separately, encoding each one, and then using a feature interaction matrix, it is possible to achieve high accuracy in the detection of hateful semantics, since both images and text are required to classify a hateful meme. We can also utilize pre-trained vision models; Chen and Pan [2] used a Transformer-based Vision-Language Pre-Training Model attached to a Random Forest Classifier to achieve an AUROC score of 76.8.

2.1. Multimodal Hate Detection and Meme Classification

Multimodal hate speech detection, especially in memes, has grown substantially since the release of the Facebook Hateful Memes dataset by Kiela et al. [6]. This dataset emphasizes the importance of joint reasoning over images and text, demonstrating limitations of unimodal and naive fusion models. Early studies have shown that simple classifiers struggle with nuanced multimodal hate [15, 5]. More recent work extends detection to fine-grained tasks such as hate severity and subtype classification [12, 8], illustrating the complexity of hateful meme analysis.

2.2. Vision-Language Pretraining and Fusion

Advances in vision-language pretraining have significantly impacted hateful meme classification. Models like ViLBERT [11], VisualBERT [10], and UNITER [3] use different fusion strategies such as early, late, or intermediate to integrate modalities. Single-stream models (e.g., VisualBERT) integrate image and text early in the pipeline, enabling fine-grained interactions. In contrast, dual-stream models like ViLBERT employ intermediate fusion via cross-modal attention. While late fusion is simpler, it typically underperforms in tasks requiring subtle multimodal reasoning [6]. Our model employs an efficient intermediate fusion, concatenating normalized embeddings from CLIP's encoders and processing them with attention-enabled classification layers.

2.3. CLIP-based Approaches

OpenAI's CLIP [13] provides robust multimodal embeddings trained via contrastive learning, proving effective for hateful meme detection. Notably, Hate-CLIPper [7] achieved state-of-the-art performance (AUROC 85.8%) on the Hateful Memes dataset by modeling explicit cross-modal feature interactions. Other works have explored prompt-based CLIP models or lightweight fine-tuning to maintain generalization [1, 14]. Our approach similarly leverages CLIP, but with simpler fusion and additional multitask heads for severity and subtype prediction.

2.4. BLIP-based Approaches

More recent vision-language models like BLIP (Bootstrapped Language-Image Pretraining) [9] have increased functionalities of multimodal representation learning by combining contrastive learning with vision-to-language and language-to-vision generation. Unlike CLIP described above, BLIP optimizes both discriminative and generative tasks, allowing for more cross-modal reasoning. This allows BLIP to better model subtle semantic interactions, which is useful for nuanced tasks like hateful meme detection. Xu et al. [17] fine-tuned BLIP on the Hateful Memes dataset, demonstrating improved AUROC compared to baseline contrastive models. This suggests that BLIP's enhanced contextual understanding helps capture sarcasm, irony, and biases.

2.5. Interpretability and Attention Mechanisms

Interpretability is crucial for multimodal hate detection due to the sensitivity of the task. Transformer-based models inherently provide interpretability through attention visualizations [11, 16]. Hate-CLIPper's interaction matrix further clarifies feature-level contributions [7]. Additionally, identifying hate targets or subtypes improves transparency [12]. Our method employs a straightforward attention gating mechanism post-encoding, offering insights into the modality (image or text) that primarily contributes to the prediction.

3. Method / Approach

3.1. Baseline Methods

The baseline methods use a multimodal model that combines two pretrained models, ResNet18 and BERT, and combines them to achieve a binary classification of hateful or not. BERT tokenizes the text with a maximum length of 64, and ResNet18 extracts visual features from the images. These two models are combined into a singular classification model which consists of a Linear Layer, ReLU, Dropout Layer, and finally a linear layer that outputs the classification. The baseline uses a learning rate of 0.0001 with the Adam Optimizer, a batch size of 16, 5 epochs, and binary cross-entropy to calculate the loss. Besides the loss, the baseline model uses several metrics to evaluate its performance. Most importantly, the AUC-ROC and F1 score signify the model's ability to classify inputs, and are a good judge of its accuracy. This, combined with analysis of the training/validation loss, provides a good picture of the baseline model's performance, making it easy to compare it to the model that our group develops.

3.2. Final Approach Overview

We implemented a transfer-learning strategy built on **BLIP** as our backbone. BLIP jointly encodes images (Vision Transformer) and text (Transformer decoder) with cross-attention, providing strong, pre-trained multimodal representations. Fine-tuning BLIP on the Hateful Memes dataset delivers three benefits:

1. **Multimodal Fusion**: BLIP combines the visual and text information all the way through the network, instead of combining the two streams together at the very end.

2. **Leveraging Scale**: Because BLIP has already been trained on huge image-text datasets, we can fine-tune it easily and with far less data to capture the intricacies of hateful memes.

3. **State-of-the-art Baseline**: BLIP is one of the most capable open-source multimodal models available, so we can focus on task-specific improvements rather than reinventing a super complex model.

3.3. Data Loading and Representation

For all experiments, we used the Facebook AI's *Hateful Memes* dataset which pairs each meme image with a short caption and stores the metadata in JSONL files. During training and evaluation the data are accessed through a custom PyTorch Dataset. For each meme, we first open the picture, resize it, and scale the pixel values to the dimensions that BLIP was trained to expect. Then, we turn the caption into tokens using the BLIP tokenizer, and add padding if necessary and an attention mask to help the model know what parts are real words.

3.4. Labeling Techniques

We experimented with different ways to add extra labels beyond the ones provided in the dataset to give the model we trained with more information. First, we tried to assign a dummy label to every meme, which led a random metrics. We then tried to create clusters from the BLIP embeddings but this did not help. We also tried to check if any words in a hate speech dictionary were in the meme but this also did not work well. Then, we tried generating extra labels with Google's BART model by asking it for details like severity and type of hate which led to be the dataset becoming larger and this led to increased recall. The final technique we decided on using is to pre-train BLIP on the labels from BART with augmentations to the images, and then fine-tune the model on the training set from the Hateful Memes Dataset.

3.5. Model Architecture

Our implementation takes advantage of the blip-image-captioning-base checkpoint and use it as the backbone to keep all pre-trained parameters intact. The pooled multimodal embedding produced by BLIP is then inputted into a single, task-specific linear layer that converts this 768-dimensional vector into one logit, which is then sent through a sigmoid function during loss computation to compute the hateful/non-hateful probability. To make implementation and debugging easier, the backbone and the classification layer are combined into a HatefulMemesDetector wrapper class.

3.6. Alternative Considered Architectures

Before deciding on BLIP, we tested two other options. First, we tried using CLIP which has separate vision and text encoders that are then combined through contrastive learning. We froze both encoders and added a small MLP but did not achieve any satisfactory results which we believe is due to CLIP blending the two modalities only after they are processed independently and this leads to the relationship between the image and text to be lost such as sarcasm linked to a particular facial expression. Another approach we tried is pairing a ResNet-50 image encoder with a BERT text encoder and concatenated their final embeddings before a classifier head. This ResNet + BERT pipeline showed the same late-fusion weakness as CLIP while bringing extra parameters and training time and worse results. BLIP was able capture these relationships with its combined modality processing, and therefore outperformed both other approaches despite requiring slightly more computation.

3.7. Training Setup

We train the model using a loss function that combines a sigmoid activation and binary-cross-entropy using PyTorch's `BCEWithLogitsLoss`. Weights are updated through the `AdamW` optimizer that adds a weight-decay of 10^{-2} to keep the weights from growing too large. The learning-rate starts low and then rises during warm-up phase of training, and then smoothly decreases based on a cosine curve. Every training step uses a batch of 32 memes and runs in mixed precision so most matrix multiplication is done in `float16` which makes training faster and lighter on GPU memory to allow us to run these models on a single GPU (usually an A100) on Google Colab. Because the hateful memes dataset we are using is not that large, we use `WeightedRandomSampler` to make sure their are roughly the same number of hateful and non-hateful memes in each batch. To help the model generalize, we augmented some images by randomly flipping them or slightly changing their colors or brightness through the Albumentations library.

4. Data

For this project, we used the Hateful Memes dataset, created by Facebook AI Research (now Meta AI). The dataset consists of 10,000 memes, split into an 8,500-sample training set, a 500-sample development set, and a 1,000-sample testing set. Each set is a JSON file, where each sample has an ID, a path to the image file, the text in the meme, and a label of whether it is hateful or not. The meme text is provided in the JSON file, so there is no need to perform OCR on the image to get the meme text. The data is comprised of the following distribution: 40% multimodal hate, 10% unimodal hate, 20% benign text confounder, 20% benign image confounder, and 10% random non-hateful [6].

The dataset was made by Facebook AI for a competition, in which participants would try to create the best-performing model for detecting hateful multimodal content. Each meme was constructed by Facebook, using images provided in partnership with Getty Images. The memes are then labeled by third-party annotators, and the ones deemed to be hateful are used to create benign confounders. Facebook defines hate speech as an attack on people based on characteristics, such as race, religion, sex, disability, and several other categories [6]. The annotators go through a 4-hour training on recognizing hate speech using Facebook's definition, before moving on to annotating the created memes. Facebook AI released the dataset via DrivenData for their competition in 2020, which was open for anyone to join. Upon the conclusion of the competition, the

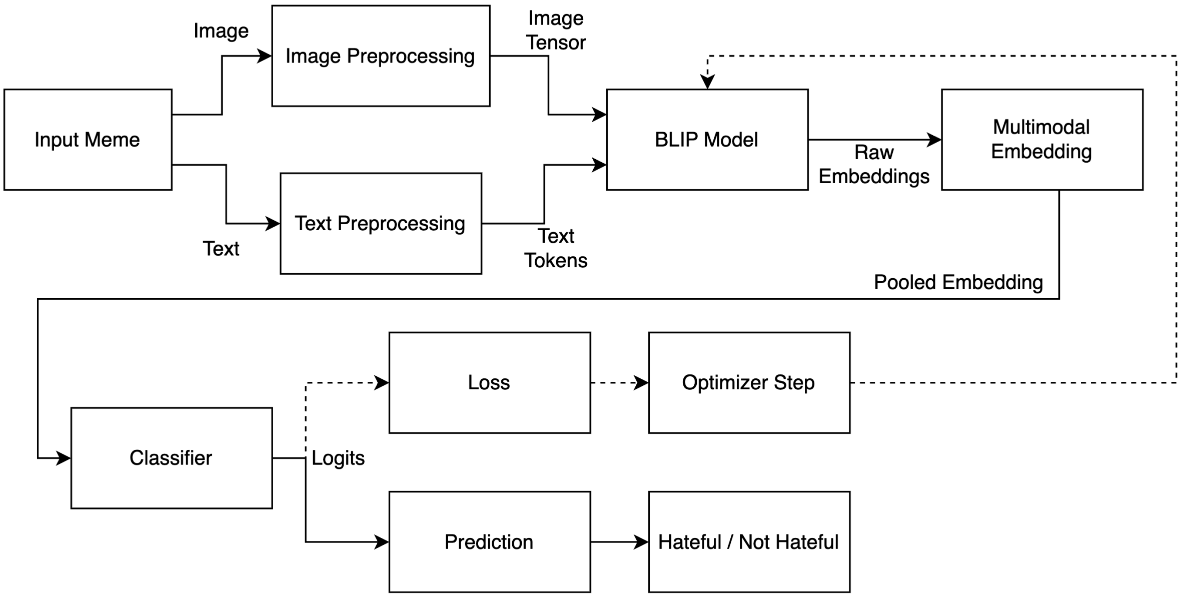

Figure 1. The model architecture for our final implementation.

dataset was posted on Kaggle, which is where we downloaded it from.

5. Experiments and Results

5.1. Evaluation Metrics:

Given the complicated nature of hateful classification, several metrics were used to evaluate the model. First and foremost, the training loss and validation loss were used in order to monitor training progress. It gave a clear indicator of whether the architecture used was viable, since in unsuccessful experiments, the training loss gradually decreased yet the validation loss remained high. This indicated that the model was overfitting or not training properly, since it was not able to correctly classify inputs as hateful or not after training. Besides those metrics, precision was used to evaluate whether the model's predictions were correct or not. Precision tracked whether the memes that the model predicted to be hateful were actually hateful according to the test set, which overall helps to deduce whether the model is outputting false positives.

One of the most important metrics used was recall. Recall denotes which of the hateful memes were classified as hateful by the model, which is slightly different from the precision in the sense that precision measures which classifications of hateful are correct, whereas recall measures which of the hateful memes are correctly classified. This is arguably the most important metric because in hateful content detection, it is most important for the model to correctly classify as many memes as possible to prevent potentially hateful content from propagating as safe for public audiences.

In terms of more technical metrics commonly applied in classification contexts, F1-score represents the harmonic mean of the precision and the recall, essentially combining the two metrics. This is crucial in balancing both false-positives indicated by the precision and false-negatives communicated by the recall, giving us a more holistic view of the model's performance. Therefore, we aim to maximize the F1-score. Lastly, the Area Under the Curve of a Receiver Operating Characteristic (ROC-AUC) score measures the model's ability to separate detection of memes from the positive (hateful) class and the negative (non-hateful) class. The baseline for ROC-AUC is 0.5, which indicates that the model is essentially guessing between positive and negative classes in its classifications. Therefore, the aim is for the ROC-AUC to be as close as possible to 1.0, which would indicate perfect classifications and distinction between hateful and non-hateful memes.

5.2. Baseline Method:

The baseline methods use a multimodal model that combines two pretrained models, ResNet18 and BERT, and combines them to achieve a binary classification of hateful or not. BERT tokenizes the text with a maximum length of 64, and ResNet18 extracts visual features from the images. These two models are combined into a singular classification model which consists of a Linear Layer, ReLU, Dropout Layer, and finally a linear layer that outputs the classification. The baseline uses a learning rate of 0.0001 with the Adam Optimizer, a batch size of 16, 5 epochs, and binary cross-entropy to calculate the loss. Once training and validation completed, we created graphs of the loss, and saved the AUC-ROC and F1 to compare against later implementations.

The loss graph paints a clear picture of the baseline model's performance. While the training loss did steadily decrease, it finished at a relatively high value of about 0.6, and the validation loss reached its minimum after the first epoch, steadily increasing from that point onwards. The poor performance of the baseline model is corroborated by the AUC score and the F1 score. The AUC score achieved by the baseline model was 0.58, which is marginally better than randomized guessing, and the F1-score was shockingly low, a mere 0.1429 for the iteration with the highest AUC score. This can be explained by several factors. Memes consist of multimodal components, text and images, and the use of such a simple model with only 2 Linear layers suggests an inability to learn the complex text/image relationships in hateful memes, namely those that contain sarcasm and subtle context. Additionally, training on more epochs and finetuning the hyperparameters such as the learning rate could result in more effective baseline results. However, the results achieved with this model are a good indicator for future results and potential for growth with the hateful memes dataset.

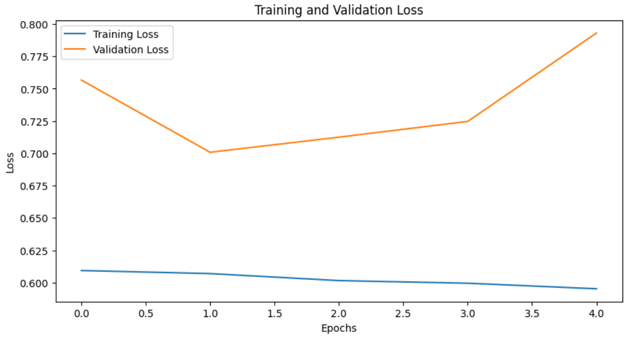

Figure 2. Training and validation metrics for the baseline BERT + ResNet18 Model over 5 epochs.

5.3. Model Experimentation

5.3.1 CLIP Baseline

This approach utilized a baseline CLIP model, utilizing a frozen CLIP with only the top 4 layers unfrozen, allowing the model to maintain its base knowledge while also allowing the model to train on the hateful memes classification task. The image and text embeddings are concatenated into a single tensor and then fed into an MLP consisting of Linear and Dropout layers with a ReLU to provide a single hatefulness classification, trained over 10 epochs. The training loss steadily decreases over time, yet the validation loss fluctuates, indicating that the model is learning the training data well, but failing to generalize it to unseen data. Additionally, while the training attains a relatively high AUC of 0.7117, as well as a precision of 0.7279, indicating that its discrimination abilities are above average, the recall is unfortunately low at only 0.3960. This suggests that although the model is making correct classifications, it is still missing a lot of positive cases. This is especially problematic in the context of detecting hate speech, since missed classifications result in hateful content slipping through filters and being seen by innocent audiences. This mix of high precision-low recall explains the F1-score of 0.5130, which reflects both categories and is exceedingly average.

5.3.2 CLIP + Dummy Labeling

Our initial approach used the OpenAI Contrastive Language-Image Pretraining (CLIP) model for hatefulness classifications. CLIP uses a text encoder, as well as an image encoder, in order to process images and provide classifications by calculating embedding similarities in a matrix. We trained the model for 5 epochs using the AdamW optimizer with a learning rate of $2 \times 10^{-}5$ and Binary Cross Entropy as the loss function for the hateful and subtype classifications, while using CrossEntropyLoss for the severity classification. Since severity and subtype labels aren't provided as part of the Hateful Memes dataset, dummy labels were assigned in order to test the model's performance. Multi-Layer Perceptrons (MLP) are used for each separate sub-task (hatefulness, severity, subtype), consisting of a linear layer, GELU activation function, and a dropout layer, with a final linear layer to produce the output. The validation performance across epochs is shown in Table 1, showing that it wasn't very good at finding the hateful memes (final F1-score: **0.34**). Increase in validation loss amidst a perpetual decrease in training loss signifies overfitting.

Table 1. Validation Metrics per Epoch for Initial CLIP-based Attempt.

Epoch	Val Accuracy	Val F1	Val AUC
1	0.5000	0.0000	0.5713
2	0.5000	0.0000	0.6296
3	0.5000	0.0000	0.6448
4	0.5500	0.3077	0.6520
5	0.5560	0.3353	0.6561

To account for this, we added regularization to the original CLIP-based classifier in the form of Dropout layers and replacing the feature interaction layer, previously consisting of just a simple Linear Layer, with a more sophisticated fusion approach, using Layer Normalization and a Dropout Layer. This proved to be more successful, achieving an Accuracy of 0.67, an F1-Score of 0.6512, and an AUC of 0.7235 after training for 15 epochs. Nonetheless, since dummy labeling was used, the results cannot be interpreted at face value, resulting in the need for more sophisticated labeling techniques to produce accurate classifications.

5.3.3 BLIP + Lexicon Heuristic

One experiment we conducted was to finetune a BLIP backbone with a simple hate-speech lexicon for ten epochs on an NVIDIA A100. We used AdamW with a learning rate of 3×10^{-4} and a weight-decay of 0.01 and a weighted BCE loss to try to make each batch balanced. Each 224 pixel image was augmented with flips and change in color. We achieved results of Validation F1 climbing from 0.05 to a peak of 0.56 with accuracy of 0.56, recall of 0.60, and AUC of 0.54 by epoch 4, after which learning plateaued and early-stopping ended training at epoch 9. This more than doubled the F1 of a naive CLIP baseline and surpassed dummy results.

To try to better the previous experiment, we tried freezing all but the final two BLIP encoder layers and added aggressive augmentations such as flipping the images horizontally and changing the brightness of the image. This training session ran for six epochs due to early stopping on an NVIDIA A100 with a AdamW optimizer with learning rate of 3×10^{-4} and a weight-decay of 0.01. This run had the best results at around epoch 6 with a recall of around 0.67, a F1 of around 0.59, and an AUC of around 0.53. In comparison to the previous lexicon attempt, recall improved by 0.07, F1 improved by 0.11, while AUC dropped by 0.01. These results indicate that the model is getting better at identifying hateful memes, but is still identifying a lot of false positives.

We tried one more time to improve upon the metrics from the previous run, and we unfroze just the last two BLIP encoder layers and applied a lighter augmentation while keeping the same hyperparameters. We trained again on an NVIDIA A100 and training stopped after six epochs due to early stopping and reached its best validation metrics at epoch 5. The metrics at this epoch were a recall of 0.89, F1 of 0.65, and an AUC of 0.52. In comparison to the previous lexicon attempt, recall improved by 0.22, F1 improved by 0.06, while AUC dropped by 0.01. These results indicate that the model got even at identifying hateful memes, but led to more false positives.

5.3.4 BLIP + LLM Labeling

Finally, we tried adding an additional model to create more detailed labels for the hateful memes. We employed a zero-shot classification pipeline using a BART model pretrained on the MultiNLI dataset, which would assign a severity label (low, medium, or high) to the text component of memes that were labeled as hateful. Non-hateful memes were assigned a low severity. These severity labels were then incorporated into our dataset for training.

We again used a BLIP base model, fine-tuning the top four transformer layers. The architecture fused the normalized image and text tokens early, feeding the combined embedding into classification heads for hatefulness and severity. We trained for 10 epochs using AdamW, a cosine learning rate scheduler with warmup, and weighted random sampling to balance the distribution of hateful/non-hateful classes within batches. The final results for this model achieved an F1 score of 0.42 and an AUC of 0.53. This performance is lower than previous implementations we've tried, suggesting that the zero-shot generated severity labels are not as good as using a lexicon heuristic.

Method	**Recall**	**F1**	**AUC**
BERT + ResNet18	0.22	0.491	0.585
CLIP Baseline	0.396	0.513	0.712
CLIP + Dummy Labels	0.546	0.651	0.724
BLIP + Lexicon	0.888	0.650	0.521
BLIP + LLM Labeling	0.364	0.419	0.532

Table 2. Validation performance of all methods evaluated.

6. Conclusion

In this project, we experimented with a variety of multimodal models to detect hateful memes, focusing on improving recall to minimize the likelihood of harmful content slipping through. Although our final BLIP-based model achieved a recall of 0.89 and an F1-score of 0.65, our best AUC remained relatively low at 0.52, falling below our original goal of surpassing 0.8. This hindsight is most likely the result of the natural difficulty of hateful meme detection, where subtle interplay between image and text requires fin-tuned semantic reasoning. Howere, achieving a high recall was ultimately the most important for our application. In content moderation, missing hateful memes (false negatives) is far more damaging than occasionally flagging benign memes (false positives). High recall ensures that the majority of harmful content is identified and can be further reviewed, even if precision is somewhat sacrificed.

Our experiments further revealed that early fusion of image and text modalities, as implemented in BLIP, was essential for modeling the nuanced semantics of hateful memes. Simpler models like ResNet+BERT or frozen CLIP struggled to capture these interactions, leading to poor recall and generalization. Attempts to improve performance through auxiliary labeling with zero-shot large language models did not succeed, suggesting that automatically generated labels lacked the necessary nuance for training better classifiers. In the future, manually annotating memes with more detailed severity levels and hate subtypes could provide richer supervision signals, allowing models to better understand the spectrum of hatefulness rather than relying solely on binary classification. Additional future work could explore scaling to larger vision-language models such as BLIP-2 or adding interpretability techniques like attention visualization to better understand and audit model decisions.

7. Discussion on Deep Learning Aspects

Because we are trying to tackle the challenge of identifying multimodal hate in memes, we employed a deep learning approach with an earlier fusion of features, compared to the late-fusion baseline approach. We leveraged transfer learning, only fine-tuning the top layers of the pre-trained BLIP model while keeping the other layers frozen. Input images were normalized into tensors, and the input text was tokenized, adhering to BLIP's expected input formats. We used Binary Cross Entropy loss, as classifying memes as hateful or not is a binary classification task. Additionally, this loss function is appropriate because our final output is a logit, so we used BCEWithLogitsLoss for our training.

Overfitting was monitored through metrics such as precision, recall, F1 score, and AUC. Generalization remains to be a challenge, as indicated by our modest AUC scores. Hyperparameters such as learning rate and weight decay were tuned empirically, and we used the AdamW optimizer and a cosine learning rate schedule. We used AdamW, a decoupled weight decay variant of the popular Adam optimizer, because of improved generalization and regularization for transformers, and we used a cosine learning rate schedule to provide a smooth, gradual reduction in learning rate. These implementations were created using PyTorch, HuggingFace, and a variety of pre-trained models such as CLIP, BLIP, and BART, providing a well-established starting point for tackling this multimodal challenge.

8. Team Contributions

Student Name	Contributed Aspects	Details
Vedesh Yadlapalli	Model Implementation, Experimentation Analysis	Implemented LLM labeling and BLIP
Michael Osmolovskiy	Baseline Model Implementation, Experimentation Analysis	Implemented and evaluated baseline ResNet + BERT
Justin Xia	CLIP Implementation, Experimentation Analysis	Implemented CLIP baseline without labeling
Soham Samal	Labeling Model Implementation and Experimentation Analysis	Added dummy labeling to CLIP model and experimented with Lexicon

Table 3. Contributions of team members.

References

[1] Jing Cao, Ming Chen, Haiyun Jiang, Lemao Ma, and Jia Liu. Prompthate: A prompt-based pre-trained model for hate speech detection, 2023. 2

[2] Yuting Chen and Feng Pan. Multimodal detection of hateful memes by applying a vision-language pre-training model. *PLoS ONE*, 17(9):e0274300, 2022. 1

[3] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[4] Arkadeep Das, Animesh K. Kolya, and Asif Ekbal. Effective techniques for multimodal data fusion: A comprehensive study. *IEEE Transactions on Multimedia*, 25:3217–3231, 2023. 1

[5] Raul Gomez, Joan Gibert, Lluis Gomez, and Dimosthenis Karatzas. Exploring hate speech detection in multimodal publications. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1470–1478, 2020. 1

[6] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes. In *Advances in Neural Information Processing Systems*, volume 33, pages 2611–2624, 2020. 1, 2, 3

[7] Gokul Karthik Kumar. Hate-CLIPper: Multimodal Hateful Meme Classification based on Cross-modal Interaction of CLIP Features. In *Proceedings of the Second Workshop on NLP for Positive Impact (NLP4PI)*, pages 171–183. Association for Computational Linguistics, 2022. Affiliation: Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI). 1, 2

[8] Kyuhwan Lee, Xi Chen, Luowei Chen, Zhe Lin, and Mu Wu. Weakly supervised vision-and-language pre-training with semantic alignment loss, 2021. 1

[9] Junnan Li, Zhaowei Fang, Caiming Xiong, and Steven C.H. Hoi. Blip: Bootstrapped language-image pretraining for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 12888–12900. PMLR, 2022. 2

[10] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language, 2019. 2

[11] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 2

[12] Shrayan Pramanick, Shivam Roy, Rituparna Singh, and Md Shad Akhtar. Detecting severity and subtype of hateful memes. In *Proceedings of the 11th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 236–241, Online, Apr. 2021. Association for Computational Linguistics. 1, 2

[13] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 1, 2

[14] Umang Shah, Gokul Karthik Kumar, Sahil Aggarwal, and Anish Mittal. Clip for all: Exploring the impact of embeddings, prompts, and training strategies on multimodal hateful meme detection, 2023. 2

[15] Shardul Suryawanshi, Aniket Mishra, Revant Marreddy, Pranit Sahoo, Mithun Kumar, and Md Shad Akhtar. Multimodal hateful meme detection: A survey, 2020. 1

[16] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, 2019. 1, 2

[17] Meng Xu, Rui Wang, and Hao Chen. Fine-tuning blip for multimodal hate detection in memes. *arXiv preprint*, abs/2304.00000, 2023. 2