The Hateful Memes Challenge: Detecting Hate Speech in Multi-Modal Memes

Narine Marutyan, Alissa Jouljian

American University of Armenia
Yerevan, Armenia

May 9, 2024

Abstract

In today's digitalized world, protecting people from the vulnerabilities associated with online hate is crucial. Detecting and addressing hate speech is essential for maintaining a safe and respectful online environment. So far, many machine learning models have been developed to identify hate speech for unimodal inputs, such as social media post captions, comments, statuses, and more. However, the challenge gets even more intense when the input becomes multimodal. It is crucial to recognize the importance of solving this issue since the world itself is not unimodal, and everything we see depends on grasping the content not unimodally but understanding the whole meaning. We decided to tackle this issue by introducing a method based on vision-text models to solve the Hateful Memes Challenge [KFM+20], which is about classifying the memes into two classes: hateful and non-hateful. Understanding the multimodal interpretation is essential for solving the challenge of hateful memes. This is because the images or texts on the memes can be harmless on their own, but they can be hateful together. This project aims to investigate and develop effective methods for detecting hate speech in multimodal memes. All the codes for the projects can be found at github.com/narinemarutyan/Hateful-Memes-CLIP

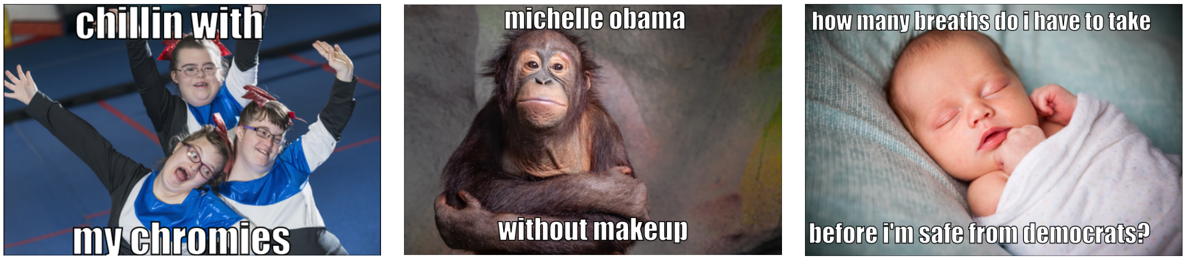
chillin with
my chromies
michelle obama
without makeup
how many breaths do i have to take
before i'm safe from democrats?

Figure 1: Demonstration of the memes showing that they were chosen so that the unimodal classifiers would struggle to classify them robustly. Thus, the content should be considered multimodally to grasp the meaning holistically.

1 Introduction

Memes are typically humorous images or videos that internet users copy and spread. Understanding the nature of these memes before they spread is essential. Although memes are mostly considered harmless, some of them express hate toward specific groups of people. Thus, it is necessary to be able to detect that kind of content. This will help social media platforms address and remove hateful content and foster a safer online environment for users.

When people view memes, they don't isolate the text and the image. Instead, they derive meaning from both components as a whole. This holistic inference allows them to form a comprehensive understanding of the meme, making it possible to understand whether the meme is hateful or not.

In Figure 2, we can see that the textual and visual parts of the meme are harmless when considered separately, whereas when considered together, they are hateful. However, for machines, this task is more complex. Combining the two modalities is essential to extract meaningful inferences about the content. This project aims to explore and develop effective techniques for detecting hate speech within multimodal memes, aiming to construct a robust model capable of accurately identifying and classifying hateful content. To address this issue, Facebook organized the Hateful Memes Challenge, held at NeurIPS 2020 [KFM+20]. This competition focused on multimodal hate speech, aiming to advance research into multimodal reasoning and understanding. The dataset was made so that the models could succeed in classifying if the meme is hateful or not, only when both textual and visual contexts were combined. In this scenario, the textual and visual parts can be non-hateful but hateful together.

Our research aims to promote further research in solving multimodal classification problems. The method we chose uses CLIP's [RKH+21] image and text encoders in combination with different image captioning methods. The designed model categorizes the multimodal memes as hateful or not. It effectively combines both images and texts in making a decision.

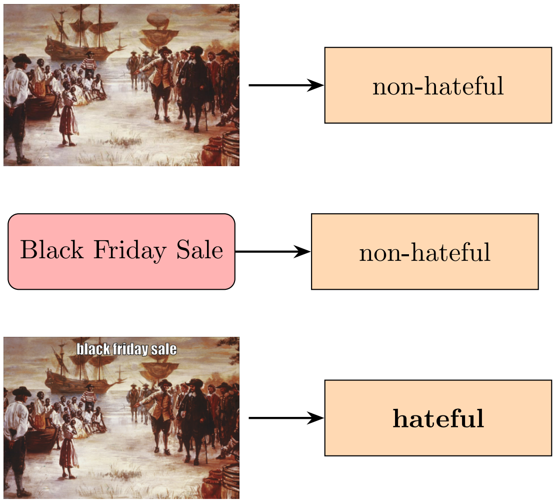

Figure 2: Figure showing how meme's both contents explicitly can be harmless, but together harmful.

2 Literature Review

Memes are rapidly evolving, and the necessity to have methods that will detect and moderate the hateful content on the internet is getting bigger. In 2020, Facebook AI announced the Hateful Memes Challenge competition [KFM+20], held at NeurIPS, and focused on multimodal hate speech. The challenge aimed to promote further research on multimodal reasoning. The competition participants tried to solve the challenge with unimodal and multimodal machine learning models. Furthermore, human evaluation was also done to understand the challenge's difficulty level. However, the results showed that surpassing human performance was not so trivial, and the best result for the competition was 84.50, which is just 2% better than human performance. Nevertheless, if we compare that result with the initial best baseline, which was 75.44, the results are not bad. Since then, much research has been done on this topic, and many exciting methods have been proposed.

A line of research in this direction includes using CLIP [RKH+21]. The idea behind CLIP is simple: mapping images and texts onto a shared embedding space. This idea has existed and even been studied for almost ten years [SGMN13]. However, only recently have the methods evolved and reached the point where they help us solve such challenging tasks as multimodal reasoning. CLIP is designed to grasp intricate relationships between textual and visual contents. It considers even small nuances and brings the contents to an embedding space where similar contents will also appear. This dual comprehension makes it possible to solve tasks like image classification, visual search, etc. Additionally, due to its versatile domain knowledge, it is easy to adapt across diverse domains and downstream tasks. To understand how this model works so well, diving deeper and understanding its architecture is crucial. CLIP consists of text and image encoders, which map high-dimensional input to low-dimensional vector spaces. Those encoders are made of Transformers. Transformers were introduced in 2017 in the paper 'Attention is All you need'[VSP+17]. That paper describes in depth how the attention mechanism works and why transformers work so well.

Recent attempts to solve multimodal reasoning tasks include using pre-trained VLMs such as Flamingo [ADL+22], LLaVA [LLWL23], GPT-4 [Ope23], etc. A paper on detecting and correcting hate speech in multimodal memes with large visual language models [VW23] utilizes a pre-trained VLM called LLaVA for detecting and correcting hateful content within multimodal memes. Since LLaVA understands both images and texts in combination, it can make decisions not unimodally but by considering both contents. Thus, for our scenario, LLaVA can utilize its capabilities of multimodal understanding to detect if the content provided is hateful or not combined with its accompanying text. Additionally, LLaVA is easily adaptable, and learning from new examples can improve its classification quality. Like many other Generative models, LLaVA offers zero-shot prompting strategies to detect hate speech. This method demonstrates a vivid shift from traditional unimodal approaches to more sophisticated multimodal ones. But here, the hardship stands in finding a good prompt to give to such models. Even though VLMs work great, training or making inferences on them demands substantial computational resources. Additionally, much effort must be put into making good prompts for those VLMs.

Some methods have been proposed to address the issues with prompt engineering for vision-language models. For instance, Context Optimization (CoOp) [ZYLL22] offers prompt learning algorithms to optimize the context given to such models. For deploying vision-language models like CLIP, CoOp offers some crucial advantages over traditional manual prompt engineering. So, the latter automates the process of prompt engineering, which would, in other cases, require an extensive period of trying and failing before ending up with a good prompt. This manual tuning is not only time and resource-consuming but can also be useless when changing the model since each model has its specificity. So, CoOp offers a small prompt learning network that can be trained to write efficient prompts instead of manual prompt engineering.

3 Methodology

To develop a machine learning model that will classify multimodal memes as hateful or non-hateful, we decided that using CLIP's encoders would be a promising idea. Using large vision language models was too expensive, and it was apparent that their performance, even without fine-tuning, would give us some results. However, since CLIP is cheap to use, easy to adapt, and promising, we decided to put our best foot forward in improving it. Contrastive Language-Image Pre-training (CLIP) [RKH+21] is a robust machine learning model which understands images in conjunction with textual descriptions. We consider solving this task with CLIP a promising work towards fostering further research on multimodal reasoning. Instead of fine-tuning CLIP we propose training a small network which will utilize CLIP's vision and text encoders to solve our classification task.

3.1 Dataset

The dataset for this multimodel reasoning task comprised image memes on which there was a text, and together, the text and image could either be hateful or not. The hardship of this problem was that, in most cases, the text or image by itself could be harmless, but when put together, it can be hateful. The dataset consisted of 10000 entries, from which 8500 belong to the test set, 500 to the dev set, and 1000 to the test set with no ground truth labels. There was a little class imbalance in the train set; out of 8500, 5450 were negative cases, meaning that the meme is harmless, and 3050 were positive cases, meaning that the meme is hateful. The dev set was not imbalanced; the classes were equal in size. To address the class imbalance issue in the train set, we utilized the two popular solutions for class imbalance problems. First, we oversampled the train set by duplicating the minority class and generating samples using the Synthetic Minority Over-sampling Technique (SMOTE) [CBHK02]. We did not achieve accuracy improvement with this algorithm and decided to undersample our dataset randomly. This worked even worse for us, as our accuracy dropped by 6 percent. Thus, we chose to stay with our original number of entries. However, one experiment we did with the dataset worth mentioning is text masking on the image. For some experiments, we decided to use OpenCV's text detection framework to mask the textual part from the memes and replace them with boxes. The results of the masking can be seen in Figure 3. This was done to understand whether image captioning improves when it does not see the text but only the visual part. Further, the results of the experiments will show what came out of it.

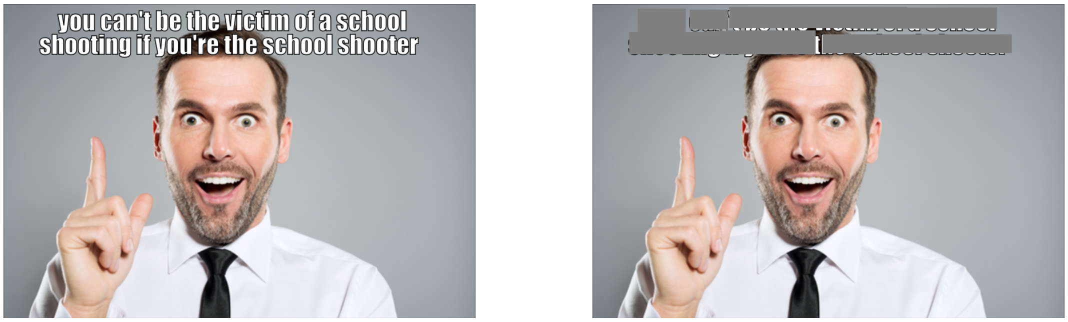
you can't be the victim of a school shooting if you're the school shooter

Figure 3: Example of masked and unmasked images.

3.2 Image Captioning

For the purpose of fine-tuning CLIP, instead of modifying CLIP's vision encoder's weights and tuning both vision and text encoders, we chose a new and authentic technique. We fed our images to image captioning models, and for each image, we got its generated caption, which describes the scene in the image. For instance, in Figure 4, you can see how the captioning model called OFA predicted the following caption for the image: 'a picture of two people shaking hands'. Since Generative models are now massively enhancing, we decided to try different models for image captioning and see how the model behaves. The first model we ran our data on was OFA [WYM+22]: a unified sequence-to-sequence pre-trained model that supports different modalities and is suitable for image captioning tasks. Next, we decided to try Bootstrapping Language-Image Pre-training (BLIP) [LLXH22]. The latter uses a technique called bootstrapped pre-training, where, at the beginning, the model learns from image text pairs to understand how texts and images are related. Training under this setting enables BLIP to perform tasks like caption generation for images based on image/text descriptions.

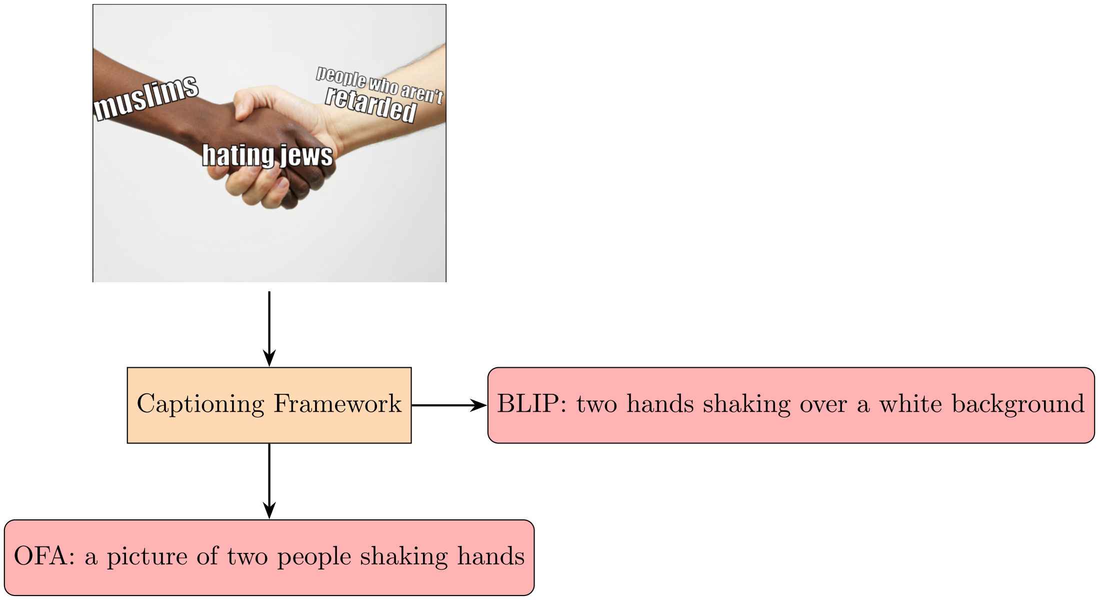
muslims
people who aren't retarded
hating jews
Captioning Framework
BLIP: two hands shaking over a white background
OFA: a picture of two people shaking hands

Figure 4: Diagram showing the results for the two image captioning models.

3.3 Our Proposed Architecture

Our proposed architecture, see Fig. 5, is as follows: Given the meme image, we perform two tasks:

1. We pass it through CLIP's image encoder to obtain an image embedding.
2. We pass the meme through our proposed captioning framework to generate a caption. During the training stage, where image text is available, we concatenate texts as follows: 'text on the meme' + [SEP] + image caption. The resulting text then passes through CLIP's text encoder to obtain the embedding for the caption and the existing text.

The obtained embeddings from the image encoder and text encoder are passed through the feature interaction map. The resulting output passes through a trainable layer added by us, which produces the final output.

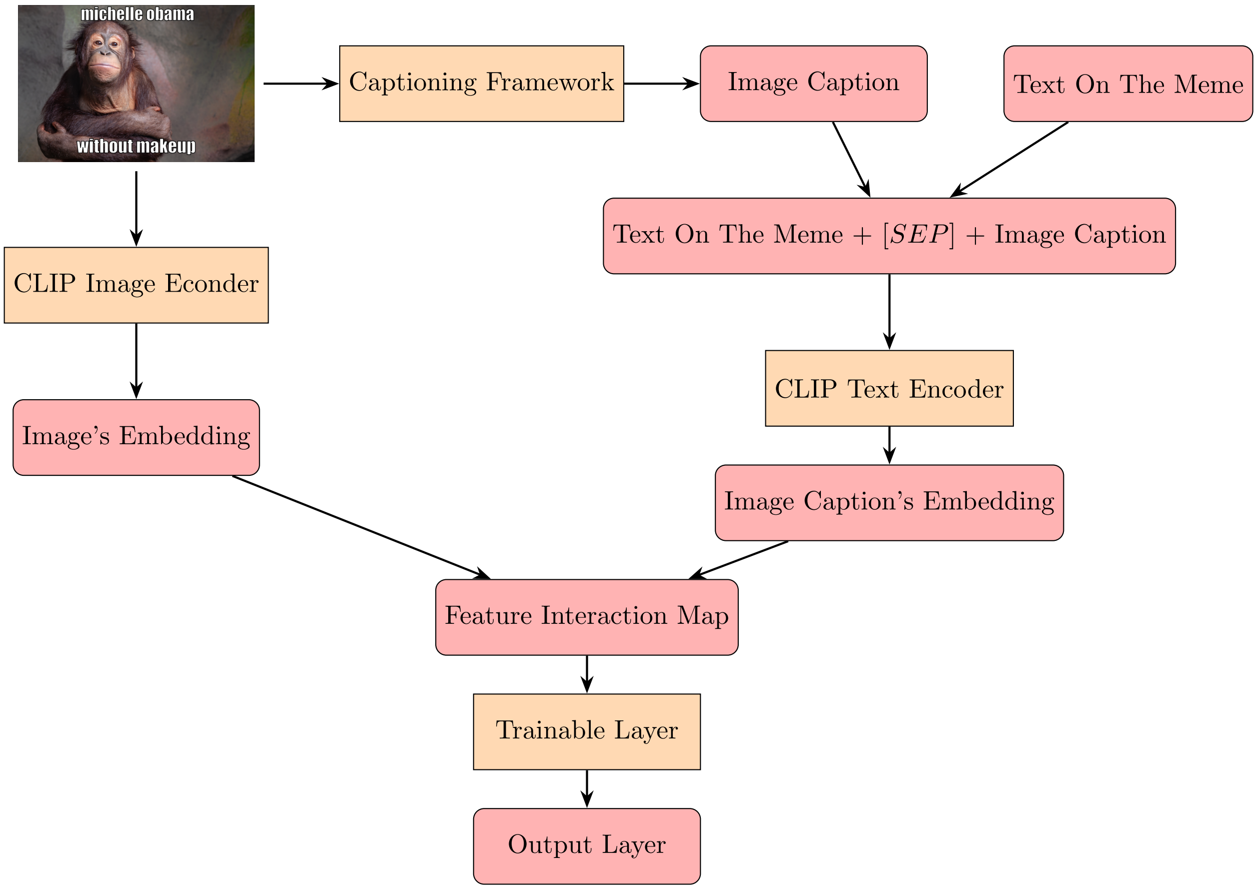

Figure 5: Diagram showing the flow of the whole pipeline

4 Experimental Results

In this section, we describe our experimental setup and summarize our results. For our experiments we used Ubuntu 22.04.4 LTS, codenamed Jammy Jellyfish with 8GB GPU memory. The hyperparameters are depicted in Table 1

4.1 Raw CLIP

Before training our network, we first tried to evaluate CLIP zero-shot with only prompt engineering by using its text and image encoders within our task. Because of its extensive training, CLIP has seen diverse image and text pairs, including images where the text is written both on the image and

Hyperparameter	Value
Image size	224
Pretrained CLIP model	ViT-Large-Patch14
Optimizer	AdamW
Maximum epochs	18
Batch size	128
Learning rate	0.0001
Weight decay	0.0001
Gradient clip value	0.1

Table 1: Hyperparameter configuration

as a caption. So, CLIP can understand and interpret the texts in the pictures. This made us think that CLIP has the potential to understand our task and classify memes as being hateful or not from zero-shot. With this approach, we found image embeddings for the memes and text embeddings for the negative and positive prompts we wrote. By finding the cosine similarity and doing softmax, we found out if the meme was closer to negative prompts, meaning being harmless, or to positive prompts, so being hateful. The results from zero-shot experiments could have been more promising. As shown in Table 2, all three prompts we used had low AUROC scores. In summary, none of the prompts attain metrics close to what we achieved via training. To elaborate more on the prompts, we first tried a self-explanatory setting of putting the negative prompt ”non-hateful” and the positive prompt ”hateful.” We made the negative prompt that way because CLIP has one serious issue: it does not process negative words such as ”not” and ”no”; it only understands negations connected with words. Thus, we connected the no part with the word. Even so, it showed dissatisfying results; the AUROC could not exceed 0.5. Therefore, we found a different approach - instead of making prompts with just general words like hateful or non-hateful, we decided to make negative prompts include all the hateful content it can consist of, such as ”Disability Discrimination”, ”Hate towards LGBTIQA+,” and more. For the positive prompt, we used words that describe nonhateful memes: such as ”Satire”, ”Humour”, and ”Sarcasm”. This improved the results by only 2 percent; it could have been better. Lastly, we tried to have multiple prompts for the third prompt, not just positive and negative. It was similar to the second prompt, but instead of combining all the Violations to one or all the types of humor in one prompt, we decided to split them into separate texts and calculate many cosine similarity scores and only then map them to a binary classification problem. With that, we got 0.51 AUROC, which is even 1% worse than the previous one. So, with zero-shot prompt engineering, we found that the results were not as good as we expected, so we moved on to the training stage.

Prompts	AUROC
Prompt 1	50%
Prompt 2	52%
Prompt 3	51%

Table 2: Raw CLIP prompt engineering results

4.2 Our Approach

After the results on raw CLIP, the decision was made to train a small network, which will be fed with several outputs received from CLIP. But before moving to CLIP, we made a change that positively impacted our network. We decided to use an image captioning network to give some description about the scene in the image for the CLIP text encoder to better grasp the content. Instead of using just one, we decided to experiment with two different image captioning models. First, we captioned our original dataset with OFA, a unified sequence-to-sequence pre-trained model. Then, we applied masking techniques using OpenCV and captioned these masked images with the same OFA model. This was done to understand the impact of masking on the generated captions. Secondly, we did the same with the BLIP model. After this, we have CSVs containing captions generated by different models for various image types. We also have the texts on the memes, which came with the dataset. Lastly, we have two folders of images, one masked and one original. Now, we are ready to train our network and get some results. After some experiments, we noticed that the network did not demand much computational resources. Training on an 8GB GPU required only 18 epochs to converge. The batch size was set to 128, which is significant for vision-text models. We experimented with two different structures to improve CLIP's training procedure. In the first method, we used two inputs: textual descriptions and image captions. We encoded the text and (text + [SEP] + caption) using CLIP's text encoder. After that, we multiplied the two encodings element by element and started the training process. However, the results did not seem as good as expected. For the second architecture, we decided to have three inputs instead of two: text, captions, and images. This required using different text and image encoders. While image encoding was done directly on the image pixel values, text encoding was done on the combination of text and captions (Text + [SEP] + Caption). As we can see in the Table 3 and Figure 6, these showed better results. Our best model was trained using images, texts and captions, where captions were done with BLIP on masked set.

Trainig Method	**Captioning Method**	**AUROC Score**
Text+Caption	OFA Original	0.68
Image+Text+Caption	OFA Original	0.7
Text+Caption	OFA Masked	0.69
Image+Text+Caption	OFA Masked	0.7
Text+Caption	BLIP Original	0.69
Image+Text+Caption	BLIP Original	0.7
Text+Caption	BLIP Masked	0.68
Image+Text+Caption	BLIP Masked	0.73

Table 3: Final Results for each method

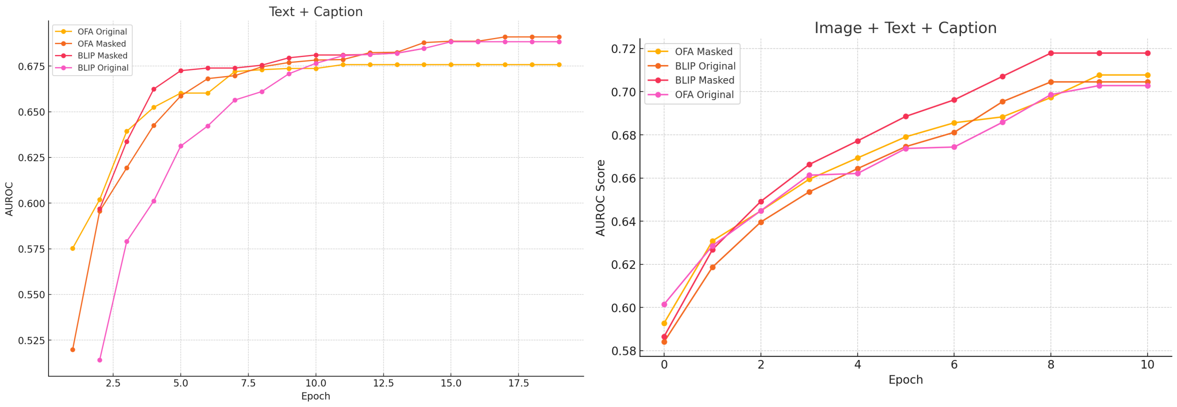

Figure 6: Visualizing the AUROC after each epoch

5 Conclusion

In this work, we proposed a simple end-to-end architecture that uses CLIP's cross-modal representations to classify memes for being hateful or not. We achieve our results with relatively small computational resources and in a very short time. With only 18 epochs, we get a 23% gain compared to Raw CLIP. The results demonstrated the effectiveness of the proposed approach, achieving an AUROC score of up to 0.73. While not cutting-edge, our work encourages further exploration in this domain as the need for multimodal reasoning grows.

References

[ADL$^+$22] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[CBHK02] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, 16:321–357, 2002.

[KFM$^+$20] 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 Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[LLWL23] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[LLXH22] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR, 2022.

[Ope23] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023.

[RKH$^+$21] 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 Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.

[SGMN13] Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. Zero-shot learning through cross-modal transfer. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 935–943, 2013.

[VSP$^+$17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[VW23] Minh-Hao Van and Xintao Wu. Detecting and correcting hate speech in multimodal memes with large visual language model. *CoRR*, abs/2311.06737, 2023.

[WYM+22] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. OFA: unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 23318–23340. PMLR, 2022.

[ZYLL22] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *Int. J. Comput. Vis.*, 130(9):2337–2348, 2022.