Harm Detection in Memes

Project report submitted
in partial fulfillment of the requirement for the degree of

Bachelor of Technology

By

Maitreyi (20bcs083)
Yash Nikam (20bcs093)
Om Morendha (20bcs095)
Samuel Mathew (20bcs116)

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
DHARWAD

CERTIFICATE

It is certified that the work contained in the project report titled “Harm Detection in Memes” by “Maitreyi (Roll No: 20bcs083)”, “Yash Nikam (Roll No: 20bcs093)”, “Om Morendha (Roll No: 20bcs095)” and “Samuel Mathew (Roll No: 20bcs116)” has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor(s)
Sunil Saumya
Data Science and Artificial Intelligence
(November, 2023)

Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Maitreyi
20bcs083

Yash Nikam
20bcs093

Om Morendha
20bcs095

Samuel Mathew
20bcs116

Approval Sheet

This project report entitled “Harm Detection in Memes” by Maitreyi, Yash Nikam, Om Morendha and Samuel Mathew is approved for the degree of Bachelor of Technology in Computer Science and Engineering.

Supervisor (s)

_____

_____

_____

Head of Department

_____

Examiners

_____

_____

_____

Date :_____
Place:_____

Contents

List of Figures

1 Introduction

With the expeditious rise of social media and its influence on society and culture, memes have become an everyday part of a person's internet consumption. The contemporary definition of a meme is a humorous piece of text, image, GIF, or video, that is spread via social media on the internet. While intended to be amusing, memes can also be pernicious, as they have the potential to spread harmful rhetoric or ideas. Given that social media is primarily dominated by young people, particularly teenagers and those in their early twenties, the presence of harmful memes poses a significant concern, as they have the potential to adversely impact the mindset and perspectives of this impressionable demographic. Such memes can even hurt the reputation of companies, individuals, eminent celebrities, and political or social groups, such as minorities.

As previously mentioned, memes might appear in various formats, but our focus will remain on image memes. Image memes generally consist of an image, which is the visual component of the meme, and some text embedded in it, which might be a caption to provide context to the image or to complement it. The combination of the two conveys a humorous message to the viewer. Hence, we say that image memes are multimodal entities, consisting of two modalities – image and text. Their multimodal nature poses a challenge to the classification of memes as harmful, as a provided piece of text may or may not be harmful given the context of the image. Thus, we can't analyze the text and image of a meme separately and we need an early fusion approach to get better results for our classification.

In some cases, it might not suffice to use the image and its text, as they may not have a direct correlation. Early fusion techniques might also fail for such occurrences. This may lead to a need for additional context to fully understand the meme's message or intent. To classify such memes accurately, it becomes essential to consider local semantics or detect entities within the image itself. By analyzing the visual elements, expressions, and objects present in the image, alongside the text, it becomes possible to gain a more comprehensive understanding of the meme's context and potential impact.

Taking all of the above into account, we aim to create a framework for the detection and classification of harmful memes to contribute to making social media spaces safer. The following outline our goals:

- Develop an end-to-end multimodal framework to analyze an input image and make predictions concerning its harmfulness.
- Train and test our framework on the HarMeme dataset[1], which has a collection of memes related to COVID-19 (Harm-C), as well as the US Presidential Elections of 2020 (Harm-P).
- Build upon the existing, best-performing multimodal framework for the above dataset, MOMENTA[2], proposed by Pramanick et al., by integrating various State-of-the-Art (SOTA) Computer Vision and Natural Language Processing models and conducting rigorous experiments.
- Exhibit that our model outperforms the benchmark we have used, MOMENTA, in terms of accuracy by 3.7 points absolute.

2 Review of Literature

2.1 MOMENTA Architecture

There have been many isolated studies in detection of hate speech , misinformation and offensive content. Given the multimodality of the memes it is necessary that the images and text be considered together to identify whether a meme is harmful or not. To achieve this Shraman Pramanic et. al. came up with the MOMENTA (MultimOdal framework for detecting hateful MemEs aNd Their tArgets) Architecture. Under this architecture they incorporated fusion techniques along with state of the art CLIP (Contrastive Language Image Pre-Training) to not only identify whether the meme is harmful or not but also determine the target of the meme. To understand the entire meaning of a meme it is important to understand both the image and the text as well. To facilitate this process the CLIP architecture was used to generate encodings for both the image as well as text. The architecture uses two kinds of fusion techniques to generate embeddings for an image. An intra-modality fusion technique and a cross modality fusion technique. There are two types of intra-modality fusion techniques designed, one for image and the other for the text.

The image intra-modality takes the encoded image representation from CLIP and the self-attended embeddings from various proposals in the meme. These proposals help identify the local context of the meme. The authors used the Google Vision API to identify different entities in the meme. These proposals were passed through the VGG model to get representations which were then passed into the image modality.

The text intra-modality takes the encoded text representation from the CLIP and the self attended embeddings from the various attributes of the image. These attributes help capture the local meaning of the text. Together these modalities capture the semantic meaning of the meme using the global and the local features given the background context.

The cross modality fusion takes the outputs of both modalities and fuses them together by assigning them different weights. Thus it is able to determine which modality should be given more importance to obtain the best results. The embedding obtained from this fusion is passed through different fully connected neural networks to determine the outputs.

2.2 Computer Vision Models

For the purpose of our project we decided to use multiple state of the art Computer vision models.

Published in 2014 by the Visual Geometry Group at the University of Oxford, the VGG model is known for its simplicity and depth. The original paper[3],explores the impact of increasing the depth of convolutional neural networks (CNNs) on image classification tasks.

VGG demonstrated that deeper networks with small receptive fields and simple convolutional layers could achieve higher accuracy in image recognition.

The YOLO model revolutionized object detection by framing it as a regression problem to spatially separated bounding boxes and class probabilities. It emphasized real-time processing and higher accuracy compared to traditional methods[4]. YOLO divides an image into a grid and predicts bounding boxes and class probabilities directly, making it efficient for real-time applications.

DenseNet introduced the idea of dense connections between layers, enabling feature reuse and alleviating the vanishing gradient problem. This architecture encourages feature propagation and enhances model compactness[5].

ResNet addressed the challenges of training very deep networks. It introduced residual connections, allowing the direct flow of information through shortcuts, making it easier to train extremely deep networks by mitigating the vanishing gradient problem[6].

The Vision Transformer applies transformer architectures to image classification. Departing from conventional convolutional structures, ViT divides an image into fixed-size patches, linearly embeds them, and processes them with transformers. This approach demonstrated the versatility of transformers beyond natural language processing, achieving competitive performance in image classification tasks [7].

2.3 NLP Models

There have been many developments in NLP models since the publication of the transformers paper. Since then the BERT model was introduced which was then fine tuned to create various other models. We used some of those models as well as other state of the art models.

MPNet utilizes a pre-training strategy that involves masking and permuting tokens in language data. The goal is to enable the model to learn contextualized representations of words by predicting masked or permuted tokens within a given context. This method is inspired by similar pre-training techniques in NLP, where models are trained on large amounts of text data to capture linguistic patterns and contextual relationships. The use of masking and permuting tokens during pre-training allows the model to understand the syntactic and semantic relationships between words and phrases, enhancing its ability to perform downstream tasks such as text classification, sentiment analysis, or language generation[8].

DistilRoBERTa is a distilled version of RoBERTa, itself a variant of BERT (Bidirectional Encoder Representations from Transformers). It incorporated a knowledge distillation approach to compress the large-scale BERT model into a smaller and faster version, while retaining its performance on various natural language processing tasks. DistilRoBERTa serves as an efficient alternative for resource-constrained environments.

MiniLM is a model compression technique for transformers. Presented by researchers at Microsoft in 2021, MiniLM employs a deep self-attention distillation process to compress

large pre-trained transformers like BERT into a smaller, task-agnostic version. By distilling the knowledge from the original model, MiniLM aims to achieve significant reduction in model size and computational requirements while preserving performance across various natural language understanding tasks[9].

3 Report on the present investigation (Methodology)

3.1 Dataset

In order to carry out our project, we employed the HarMeme dataset. This dataset consists of two divisions: Harm-P and Harm-C. Harm-P consists of memes related to US Politics while Harm-C consists of memes related to Covid-19. The creators of the dataset did so by scraping various publicly available meme pages in addition to a keyword-based search on Google. They further hired annotators to appropriately label each of the memes in one of three categories: not harmful, partially harmful or very harmful. Moreover, each of the harmful (partially harmful and very harmful) memes are labeled with a target. The target could be either Individual, Organization, Community or Society.

We have implemented our architecture on the Harm-C dataset alone. It consists of a total of 3,544 memes. The dataset has three splits: train, test and validation. The training dataset accounts for 85%, test for 10% and validation for 5% of the total dataset. Each of these datasets consist of an image ID, the image name, its labels and the text of the meme.

3.2 Feature Extraction (Object proposals and text attributes)

The model architecture consists of three primary components: the CLIP model for image-text preprocessing, the natural language model for text processing and the computer vision model for image processing. We obtain embeddings from each of these components before running them through a fusion mechanism. The feature obtained finally from the fusion layer is then fed into a neural network which gives us the desired label. In order to obtain necessary embeddings from the language and vision model, we were required to extract more features from the meme itself than the dataset provided us.

We began by extracting several required features from Google Cloud Vision API. These included bounding boxes, entities and the best label of each image. Bounding boxes contained information about object proposals in meme images. Entities consisted of context extracted from the text on the meme. These referred to text attributes contained in a meme. The best label is essentially a caption for the image. The extraction of ‘best label’ was not previously done for this dataset. We extracted the same based on our discretion of what would improve the performance of our model.

Additionally, we also used pretrained YOLO to detect the faces in an image. These were added to the object proposals obtained from Google Cloud Vision API.

These extracted features were then fed into the model architecture, as shown in figure 1.

3.3 Model Architecture

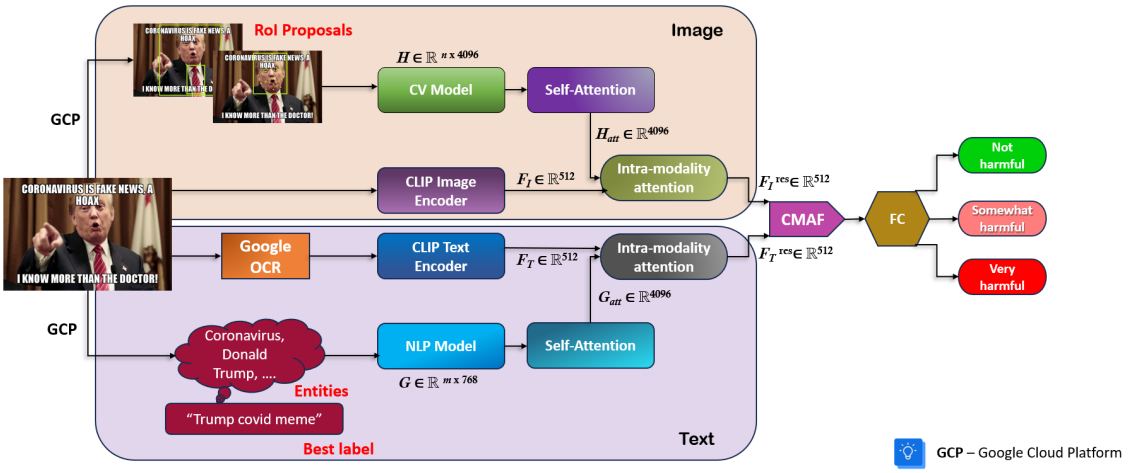

Figure 1: Complete model architecture

Each of the components depicted in figure 1 are explained in detail in the following sections.

We implemented the above model for 3-class classification as well as 2-class classification. 3-class classification was conducted by training the model on the three labels available in the dataset: not harmful, slightly harmful and very harmful. For 2-class classification, the images labeled slightly harmful and very harmful were combined into one label: harmful.

3.4 CLIP Representations

CLIP is a large language model pre-trained on a massive dataset of image-text pairs. It has excellent zero-shot capabilities, meaning it can be used to classify images without being explicitly trained on those categories.

In our model, we extract a CLIP image embedding and a CLIP text embedding for each meme. These embeddings (F_I and F_T, respectively) are 512-dimensional vectors that represent the image and text in a shared semantic space.

To obtain the CLIP image encoding, we simply pass the image itself through the CLIP model, while to obtain the CLIP text encoding, we pass both the image and its text through the model.

This allows us to learn a multimodal representation of each meme that captures both the visual and textual information. Further, we use this representation in combination with other embeddings to accurately classify memes as harmful or not harmful.

3.5 Computer Vision Model

Our computer vision models served the purpose of extracting suitable representations of the regions of interest for each image. This was achieved by feeding the extracted bounding boxes into a computer vision model. We tried and tested six newer computer vision models to see which one would improve the overall accuracy of our entire architecture. CV models we used are as follows: ResNet, DenseNet, VGG-16, VGG-19, ViT and YOLOv8.

The size of each image's embeddings, ($H \in R^{n \times 4096}$) is of dimension (n x 4096) with n being the number of object proposals for that image. These embeddings are then passed through a self attention layer which creates a representation of size 4096 of the form ($H_{att} \in R^{4096}$).

3.6 Natural Language Processing Model

The purpose of the natural language model in the architecture is to obtain suitable representations of the text features. This includes the entities as well as the best label. We used various NLP models to extract embeddings of the best label and the entities. These were then concatenated in order to retain most information.

Various NLP models used are as follows: MiniLM, DistilRoberta and MPNet.

The NLP model creates an embedding of size 768 for each of the m entities or text attributes of an image, and for the best label. All the embeddings ($G \in R^{m \times 768}$) of the entities (mx768) are passed through a self attention layer, which creates one embedding of size 768. The final embedding is then concatenated on the best label embedding of size 768. This concatenated embedding (size 1536) is passed through a linear neural layer in order to obtain the final embedding (G_{att}) of size 768.

The culminated embedding obtained from the NLP model is of the form ($G_{att} \in R^{768}$).

Processing and extraction of the image and text embeddings on the same script increased computational complexity significantly, and thus, we extracted the NLP features separately and saved it into a file. The contents of these files were then used in the main script.

3.7 Attention Fusion and Prediction

After self-attending the object proposals (H_{att}), we fused them with the CLIP image features (F_I) in an intra-modality attention module. This step combines the local image descriptions with the global semantics of the meme. Similarly, we fused the ultimate embedding obtained from the language model (G_{att}) with the CLIP text features (F_T). Overall, the local and global features capture the semantics of the meme, considering the background context. We then added a dense layer to make the dimensions of both features the same, 512.

$$F_I^{res} = W_I \otimes [F_I, Dense(H_{att})] \quad (1)$$

$$F_T^{res} = W_T \otimes [F_T, Dense(G_{att})] \quad (2)$$

Finally we fed the resulting image and text features, (F^{res}_{I}, $F^{att}_{T} \in R^{512}$) into a cross modality attention fusion layer (CMAF) to obtain the final multimodal meme representation. Some memes are more heavily reliant on text, while others are more heavily reliant on images. CMAF uses an attention mechanism to fuse the textual and visual representations, giving more weight to the modality that is more relevant to the specific meme.

$$F^{V}_{Meme} = (1 + a_{v})F^{res}_{I} \quad (3)$$

$$F^{T}_{Meme} = (1 + a_{t})F^{res}_{T} \quad (4)$$

$$F_{Meme} = W_{F} \otimes [F^{V}_{Meme}, F^{T}_{Meme}] \quad (5)$$

Finally, we fed the final multimodal meme representation F_{Meme} into two fully-connected neural layers for the final classification.

4 Results and Discussions

4.1. Metrics

We have used 3 main metrics to evaluate the several models: Accuracy, Macro-F1, and Macro-Averaged Mean Absolute Error (MMAE). For the first two, higher values are better, while for MMAE, lower values are better.

4.1.1. Accuracy

Accuracy is an evaluation metric that measures the performance of a model by taking the ratio of the correct predictions to the total number of predictions.

$$Accuracy = \frac{Correct\ Predictions}{Total\ Predictions} \tag{6}$$

4.1.2. F1 Score

F1 Score is defined as the harmonic mean of the precision and recall of a particular task, we have taken the macro average of the respective F-1 scores in the case of 3-class classification.

Mathematically,

$$Precision = \frac{True\ Positive}{False\ Positive + True\ Positive} \tag{7}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{8}$$

$$F1 = \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{9}$$

4.1.3. Macro-Average Mean Absolute Error

$$Mean\ Absolute\ Error = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{10}$$

$y_i = prediction$

$x_i = True\ Value$

$n = Total\ number\ of\ data\ points$

We then take a macro average of all the Mean Absolute Errors.

4.2. Results Table

CV	NLP	Accuracy	F1 Score	MMAE	Accuracy	F1 Score	MAE
Model		**3-Class Classification**			**2-Class Classification**		
VGG-16	MPNet	0.8079	0.5357	0.4866	0.8588	0.8454	0.1412
YOLOv8	MPNet	0.7994	0.5314	0.4874	0.8136	0.7928	0.1864
YOLOv8	DistilRoBERTa	0.7966	0.5283	0.4924	0.7966	0.7812	0.2034
VGG-19	DistilRoBERTa	0.791	0.5187	0.5431	0.8362	0.8157	0.1638
YOLOv8	MiniLM	0.7881	0.5245	0.4755	0.8023	0.7811	0.1977
Densenet	DistilRoBERTa	0.7853	0.5917	0.4737	0.8333	0.8215	0.1667
VGG-16	DistilRoBERTa	0.7825	0.5093	0.6058	0.8192	0.791	0.1808
VGG-19	MPNet	0.7797	0.5468	0.5019	0.8333	0.8226	0.1667
Densenet	MiniLM	0.7768	0.5096	0.6174	0.8107	0.7857	0.1893
ViT	MPNet	0.774	0.5201	0.4685	0.8277	0.8209	0.1723
MOMENTA		0.7710	0.5474	0.5132	0.8382	0.8280	0.1743
VGG-16	MiniLM	0.7655	0.5094	0.5116	0.8192	0.8067	0.1808
ResNET	DistilRoBERTa	0.7599	0.5027	0.5501	0.8051	0.7893	0.1949
Densenet	MPNet	0.7514	0.5561	0.4888	0.8192	0.8084	0.1808
ViT	DistilRoBERTa	0.7373	0.486	0.6006	0.7768	0.7579	0.2232
ViT	MiniLM	0.7373	0.5471	0.513	0.8023	0.7899	0.1977
VGG-19	MiniLM	0.7316	0.4952	0.4881	0.7853	0.7809	0.2147
ResNET	MiniLM	0.7175	0.5152	0.5973	0.7655	0.7456	0.2345
ResNET	MPNet	0.7034	0.47	0.547	0.7599	0.7498	0.2401
$\Delta(Best\ Model\ -\ MOMENTA)$		**3.69 %**	**-1.17 %**	**2.66 %**	**2.06 %**	**1.74 %**	**3.31 %**

Table 1: Performance on the two tasks. For two-class, we merge very harmful and partially harmful.

4.3. Discussion

After running 6 different types of Computer Vision models and 3 different types of NLP models, we observed that VGG-16 generally gave us the best results as a computer vision model and MPNet gave us the best result as an NLP Model. Therefore the combination of both of these models provided us the best results. We achieved an improvement of 3.69% i.e., from 77.10% to 80.79%, this is a quite significant improvement since the MOMENTA model improves from a previous best accuracy of 75.71% to 77.10% i.e., a 1.39% improvement. We also observed a significant improvement in MMAE, however we did not find any improvement in F1 scores. All of the CV models except DenseNet failed to classify the memes as “very harmful” due to the unbalanced nature of the dataset where only a small percentage of the memes were labeled as very harmful. But, when the experiment was run as a 2-class classification model, we observed that all metrics including F1 score had a significant improvement and showed great promise.

We had a total of 17 unique models other than VGG-16 and MPNet, and the other model that showed good promise was DenseNet and DistilRoBERTa, which was able to classify the memes as “very harmful” in spite of the unbalanced nature of the dataset and gave us a very significant improvement in F1 score, 59.17% over 54.74% in MOMENTA. YOLOv8 also provided us with a very promising result, and we observed that all three models that involved YOLOv8 consistently performed better than MOMENTA. However, the CV models ResNet and Visual Transformer, and the NLP model MiniLM did not show any improvement over the original MOMENTA model as shown in Table 1.

5 Conclusion and Future Work

We tweaked around the original implementation of MOMENTA and introduced newer and better CV and NLP models due to which we achieved a significant improvement over the original model. We also extracted the “best-label” for each image that helped us get a more in depthful insight into each image. Due to these changes, our best model outperformed MOMENTA in all baselines but one.

Although we only focused on Harm-C as our dataset for this project, we can look into working on Harm-P as future work. We would also like to look into fine-tuning the sub-models and try to implement our model to memes in video format.

6 References

1. Shraman Pramanick, Dimitar Dimitrov, Rituparna Mukherjee, Shivam Sharma, Md. Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021a. Detecting harmful memes and their targets. In *Findings of the Association for Computational Linguistics*, ACL-IJCNLP ’21, pages 2783–2796.
2. Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Md Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021b. MOMENTA: A Multimodal Framework for Detecting Harmful Memes and Their Targets. In *Findings of Empirical Methods in Natural Language Processing* (EMNLP) ‘21, arXiv:2109.05184.
3. Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition ‘14, arXiv:1409.1556
4. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi.You Only Look Once: Unified, Real-Time Object Detection ‘15, arXiv:1506.02640
5. Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger. Densely Connected Convolutional Networks ‘16, arXiv:1608.06993
6. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition ‘15, arXiv:1512.03385
7. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale ‘20, arXiv:2010.11929
8. Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, Tie-Yan Liu. MPNet: Masked and Permuted Pre-training for Language Understanding ‘20, arXiv:2004.09297
9. Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, Ming Zhou. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers ‘20, arXiv:2002.10957