# Propaganda detection in Memes

Aditya Singh Rathore
2018007
aditya18007@iiitd.ac.in

Mayuri Neeraj Mathur
PhD21033
mayunim@iiitd.ac.in

Tanmaya Gupta
2018200
tanmaya18200@iiitd.ac.in

## Abstract

*Memes have become a ubiquitous entity of all our social interactions. Be it war, elections, festivals, sporting events or a movie release, the event will most likely spread the internet through memes. Memes are extensively used for propaganda, disguised as humour, becoming a cheap and dangerous tool. It is important detect propaganda and make users aware about it so they can make informed choices and combat misinformation. We aim to detect the propaganda technique used in a meme using deep learning.*

## 1. Introduction

### 1.1. Problem Statement

In this project, we aim to use multi-modal deep learning to identify what different propaganda techniques are present in the memes, a task presented in [3].

The problem is a multi-label, multi-class classification problem.

### 1.2. Input Output

The images used by are the memes collected from Facebook from the topics such as gender equality, COVID, politics, and more. There is a total of 950 images in the dataset and the dataset is split into train set, test set, and validation set. The train set comprises 687(72%) memes, the validation set comprising of 63(7%) memes and the test set comprises 200(21%) memes.

The memes have corresponding annotations and each annotation comprises the label and the text caption of that meme. The label can be a uni-class label or a multi-class label. There are 22 unique labels that are used for the classification of propaganda techniques. 63% of the labels in the dataset belong to Smears propaganda technique, 51% of the labels in the dataset belong to Loaded Language propaganda technique, 36% of the labels in the dataset belong to Name Calling/Labeling propaganda technique and these 3 labels form the most frequent pair of labels appearing as a multi-class label. This distribution is shown in the figure 3 from [3].

A few samples from the dataset are shown in the figure 1 and figure 2. Figure 1 has 'Appeal to (Strong) Emotions', 'Appeal to fear/prejudice', 'Loaded Language', 'Slogans', 'Smears' as labels from the 22 Propaganda Techniques. Figure 2 has multi-class labels 'Exaggeration/Minimisation', 'Glittering generalities (Virtue)', 'Loaded Language', "Misrepresentation of Someone's Position (Straw Man)", 'Name calling/Labeling', 'Smears' from the 22 propaganda techniques.

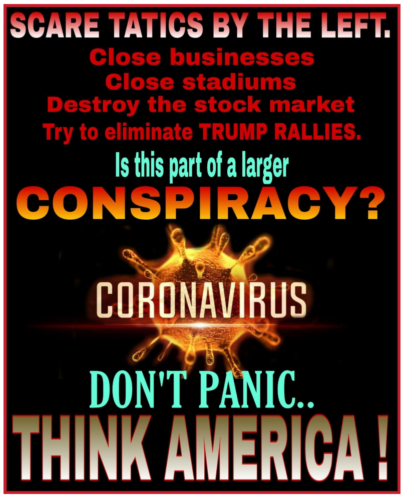

Figure 1. Meme

### 1.3. Challenges Involved

Following are the main challenges involved in the project:

- **Context of meme**: Many times, we cannot gather context of a meme from the image and text only.
- **No correlation between image and text**: Lots of memes have no correlation between the text of meme and the background image.
- **Lack of existing work**: There is no published work available as far as propaganda detection in memes is

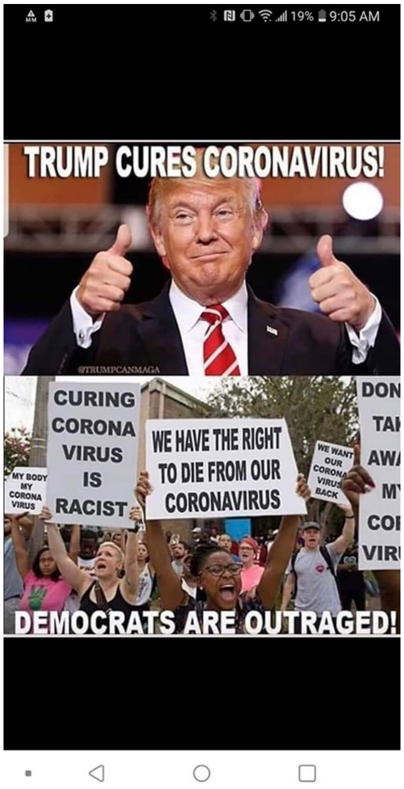

Figure 2. Meme

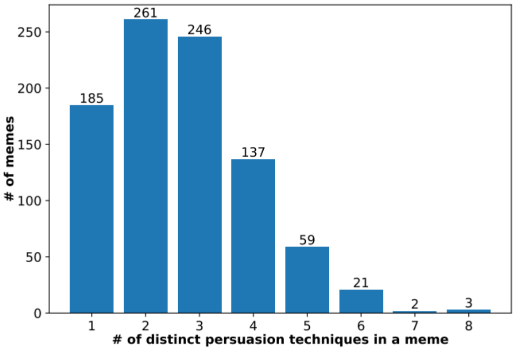

Figure 3. Propaganda Techniques Distribution [3]

considered. The closest tasks are hate speech detection and sentiment analysis.

- **Small Dataset** There are only 950 memes. The process of creating the dataset is very cumbersome as there are 22 different propaganda techniques, each labelled manually by annotators, so we cannot expand it easily.

## 1.4. Motivation

Social media has become an integral part of our daily life. On average we spend more than 60 minutes of social media website. A decade/two decades ago, most of the content on the internet was related to people's interests. There were stories about football teams, actors, etc. Story is a bit different today. The free and anonymous opportunities provided by social media has made it a hotbed of spreading propaganda. Political parties, governments, etc. have been using social media quite aggressively in recent times. We have seen in the current Ukraine war, social media access was cut for the Russian government.

An important element in new age warfare is memes. Memes are anonymous and are aggressively used to spread propaganda. Such content is dangerous, communal and comes disguised as funny/sarcasm. The effects are dangerous both in terms of mental health and persecution of targeted group.

If we are able to identify what type of propaganda is used in a meme, we would be able to provide the user a perspective on the meme. They can distinguish between what is misinformation or harmful in a better way.

## 1.5. Contributions

Our main contributions will be:

- Providing solutions to [3]

## 1.6. Proposed approach

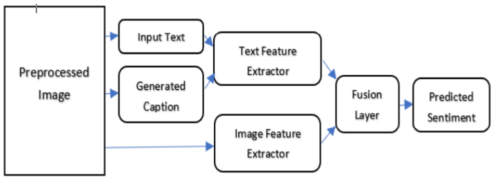

Figure 4. Architecture [6]

[6] uses inpainting to generate captions and process image and gets better accuracy with all models. The general methodology of our implementation involves applying image inpainting to the images for image reconstruction after the textual feature extraction from the image followed by passing the inpainted image to the deep neural network model used in [6] along with the corresponding text of the image as shown in figure . The model consists of :

- Image Caption Generator
- Image Feature Extractor
- Text Feature Extractor
- Fusion Layer

### 1.6.1 Image Caption Generator

Text captions corresponding to each image are converted to tokens of length 90 which are used to generate 1 - 4 words corresponding to each image of the dataset. These words are further used in text feature extraction model

### 1.6.2 Image Feature Extractor

The input image is resized to (224, 224) dimension which is passed to the image caption generator model. ResNet50 model is used for image caption generator which comprises of 50 convolutional neural network layers pretrained on ImageNet database.

### 1.6.3 Text Feature Extractor

The embedding from the image caption generator is used for extracting the context along with the meme caption to reduce bias in polarity.

### 1.6.4 Fusion Layer

This layer concatenated image feature and text feature and uses Softmax function for classification.

Our contribution involves detecting propaganda techniques on a different dataset that involves topics such as politics, covid, gender quality, and more.

# 2. Literature Survey

## 2.1. Studies on Memes

Memes have become an important means of communication in the modern social networks. Various groups have heavily deployed memes to target a person, a political party or religious and racial communities. The general anonymity associated with sharing memes has led to many people sharing objectionable content. Thus, a significant amount of research has been spent on detecting hateful and negative contents automatically using deep learning.

### 2.1.1 Classification

Facebook created a challenge [7] for detecting hate speech in memes. The dataset is designed such that unimodal methods fail. One of the prize winning solution [13] using transformer architecture visualBERT [8] and Majority Voting for classifying a meme as hateful or not. Another solution, with a higher accuracy [5] uses two different transformer architectures UNITER along with visualBERT, significantly improving the score.

### 2.1.2 Sentiment Analysis

To get a better understanding of memes, we explored multimodal sentiment analysis techniques also. [6] proposes a model which can be used for our task: text is removed from the image. A caption is generated for the image. The generated caption and OCR are combined to get the text features. Resnet50 is used for image feature extraction. Both the text and image features are combined through a fusion layer followed by a dense layer to get classification results. [4] has shown that combining image captions with OCR provides better results for meme sentiment analysis.

### 2.1.3 Meme families

A common characteristic of memes is that many of them are based on a common template like Doge. The entity and their attributes are usually same for a template family. [1] uses a multi modal deep learning model to classify an image as meme or not. It then uses image similarity, OCR and facial recognition to group memes into template families and study the families.

### 2.1.4 Propaganda detection

[3] creates a corpus of 950 memes annotated with 22 different types of propaganda techniques. Based on several state-of-the-art textual, visual, and multimodal models, they conclude that using both models for detecting propaganda performs best, based on f1-macro.

### 2.1.5 Target Detection

[10] produces dataset with degree of harmfulness and their targets with 3,544 memes. They further benchmark various unimodal and multimodal methods and conclude that multimodal methods perform better. In, [11], a neural network Momenta is proposed for detecting target of harmful memes. It combines CLIP [12] features along with detected faces, foreground objects and image attributes using Google Cloud vision api.

## 2.2. Visual Linguistic Models

On a human level, we find the target of memes by fusing information from both image and text on the meme. For example, there is a text, ”Smells amazing” and image of ”garbage”. Using text only and image only will lead to opposite conclusions. Thus, multi-modal learning” both text and image becomes an important for getting more insights from memes.

[2] uses VinVL [14] for image encoding and Oscar [9] as fusion module:along with random forest classifier for hateful meme classification.

[11] uses OpenAI's CLIP [12] to generate text and image embeddings from memes.

## 3. Propaganda Techniques

1. **Loaded language**: Using specific words and phrases with strong emotional implications (either positive or negative) to influence an audience.

2. **Name calling or labeling**: Labeling the object of the propaganda campaign as something that the target audience fears, hates, finds undesirable or loves, praises.

3. **Doubt** : Questioning the credibility of someone or something.

4. **Exaggeration / Minimisation** : Either representing something in an excessive manner: making things larger, better, worse (e.g., the best of the best, quality guaranteed) or making something seem less important or smaller than it really is (e.g., saying that an insult was actually just a joke).

5. **Appeal to fear / prejudices** : Seeking to build support for an idea by instilling anxiety and/or panic in the population towards an alternative. In some cases, the support is built based on preconceived judgements.

6. **Slogans** : A brief and striking phrase that may include labeling and stereotyping. Slogans tend to act as emotional appeals.

7. **Whataboutism** : A technique that attempts to discredit an opponent's position by charging them with hypocrisy without directly disproving their argument.

8. **Flag-waving** : Playing on strong national feeling (or to any group; e.g., race, gender, political preference) to justify or promote an action or an idea.

9. **Misrepresentation of someone's position (Straw man)** : Substituting an opponent's proposition with a similar one, which is then refuted in place of the original proposition.

10. **Causal oversimplification** : Assuming a single cause or reason when there are actually multiple causes for an issue. This includes transferring blame to one person or group of people without investigating the complexities of the issue.

11. **Appeal to authority** : Stating that a claim is true simply because a valid authority or expert on the issue said it was true, without any other supporting evidence offered. We also include here the special case where the reference is not an authority or an expert, which is referred to as Testimonial in the literature.

12. **Thought-terminating cliche** : Words or phrases that discourage critical thought and meaningful discussion about a given topic. They are typically short, generic sentences that offer seemingly simple answers to complex questions or that distract the attention away from other lines of thought.

13. **Black-and-white fallacy or dictatorship** : Presenting two alternative options as the only possibilities, when in fact more possibilities exist. As an the extreme case, tell the audience exactly what actions to take, eliminating any other possible choices (Dictatorship).

14. **Reductio ad hitlerum** : Persuading an audience to disapprove an action or an idea by suggesting that the idea is popular with groups hated in contempt by the target audience. It can refer to any person or concept with a negative connotation.

15. **Repetition** : Repeating the same message over and over again, so that the audience will eventually accept it. 16. Obfuscation, Intentional vagueness, Confusion: Using words that are deliberately not clear, so that the audience may have their own interpretations. For example, when an unclear phrase with multiple possible meanings is used within an argument and, therefore, it does not support the conclusion.

16. **Presenting irrelevant data (Red Herring)** : Introducing irrelevant material to the issue being discussed, so that everyone's attention is diverted away from the points made.

17. **Bandwagon** : Attempting to persuade the target audience to join in and take the course of action because "everyone else is taking the same action."

18. **Smears** : A smear is an effort to damage or call into question someone's reputation, by propounding negative propaganda. It can be applied to individuals or groups.

19. **Glittering generalities (Virtue)** : These are words or symbols in the value system of the target audience that produce a positive image when attached to a person or an issue.

20. **Appeal to (strong) emotions** : Using images with strong positive/negative emotional implications to influence an audience.

21. **Transfer** : Also known as association, this is a technique that evokes an emotional response by projecting positive or negative qualities (praise or blame) of a person, entity, object, or value onto another one in order to make the latter more acceptable or to discredit it.

## 4. Methodology

One of the implementations involves using a pre-trained image encoder for image feature extraction, a text encoder for extracting text embedding, and our multimodal fusion model. We used pre-trained weights of the Resnet18 image model from PyTorch and pre-trained weights of the sentence transformer from Hugging Face. The models involved are image encoder, text encoder, fusion model, and classification model. The overall architecture is shown in figure.

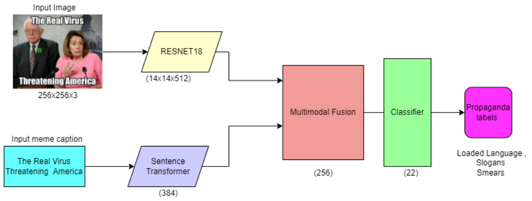

Figure 5. Baseline Architecture

#### 4.0.1 Image Encoder

For the image encoder, we have used the pre-trained weights of the Resnet18 model provided by PyTorch and extracted features only till the last convolutional layer of the Resnet18 model. During the training phase, the model weights are not updated and remain frozen during the entire training. We feed the RGB image of size $256 \times 256$ into the image encoder which gives feature size of $14 \times 14 \times 512$. This feature is further passed into the fusion model.

#### 4.0.2 Text Encoder

The meme caption is passed into the sentence transformer by HuggingFace to generate a text embedding of size $384$ for all variable lengths of the annotations corresponding to a meme image using pre-trained weights. Using sentences over word tokens helps in contextualizing the contents of the caption. This embedding is passed as input to the fusion model.

#### 4.0.3 Multimodal Fusion Model

In our implementation, the weights of the multimodal fusion model are trained and the weights gets updated after each epoch. The image feature from the image encoder and the text embedding from text encoder is passed as an input to the multimodal fusion model. The image feature is further passed through two convolutional layers each followed by batch normalization and leaky relu activation function. Then this feature is flattened and is passed through three fully connected layers with batch normalization and leaky relu activation function after each layer. This gives an image feature vector of dimension $128$. The text feature is passed through two fully connected layers followed by leaky relu activation function after each layer, which gives text embedding of dimension $128$ . These two features are concatenated to give a common feature representing both image and text of size $256 \times 256$. This concatenated feature is used as an input for the classifier for multilabels classification.

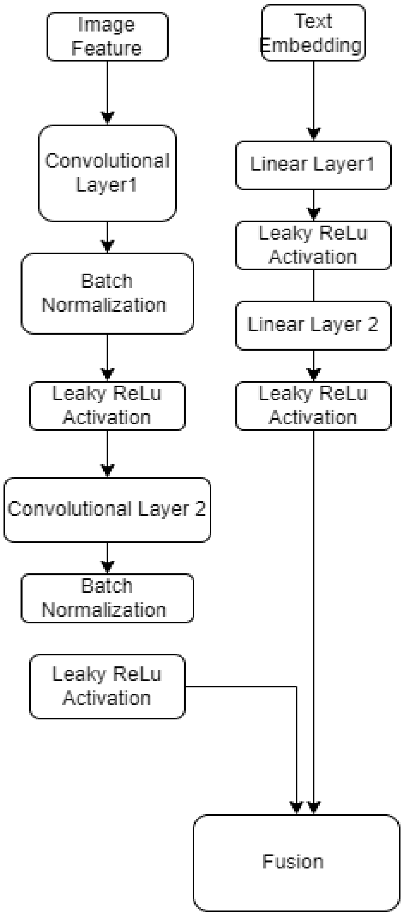

Figure 6. Multimodal Fusion Model

#### 4.0.4 Multilabel Classification Model

Multilabel classification model is used for detecting the propaganda corresponding to each image and meme caption. The concatenated feature from the fusion model is used for multilabel classification. The classifier model comprises three fully connected layers. The first layer and second layer use leaky relu activation function. The third layer is used for computing probabilities corresponding to 22 propaganda labels by using Sigmoid activation function applied on the output from this layer to fetch multiple labels.

The multimodal fusion layer and the classifier models are trained end-to-end using Binary Cross Entropy loss

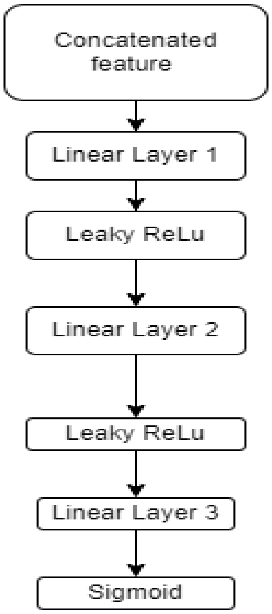

Figure 7. Multilabel Classification Model

function and Adam optimizer with learning rate of 0.001 and weight decay of 0.0001 for 450 epochs.

## 4.1. Transfer Learning

The size of our dataset is a major challenge. Our training dataset has only 687 images, Validation data has 63 images and Testing data has 200 images. There is a very high chance of over-fitting because of small size of our dataset. Another challenge that the small dataset introduces is the limited features. The model is not able to learn lots of features for classification. To tackle these problems, we explored transfer learning in our models. The general idea is

- Use a large model trained on very large datasets to extract features from memes. These features should capture the global conetxt of memes.
- Use these features on a simple multi layer perecptron network. A small network would prevent overfitting on the dataset.

We explored various combinations of following networks for generating features:

- OpenAI's CLIP: CLIP is a neural network that learns visual concepts from natural language supervision. The network is trained over 400 million image text pairs from the internet. We can get text and image encoding from CLIP. These encodings should capture the global context of a meme well.
- BERT: The context size of CLIP is limited to length 77. However, in many cases the text of a meme is much larger. Further, in many memes, text and image are not correlated. We have used a distilled version of BERT from Hugginface library to generate embeddings for the text of memes. These embeddings should capture the semantic meaning of text.

### 4.1.1 CLIP features only

We concatenated text and image features from CLIP and passed them through to a simple MLP.

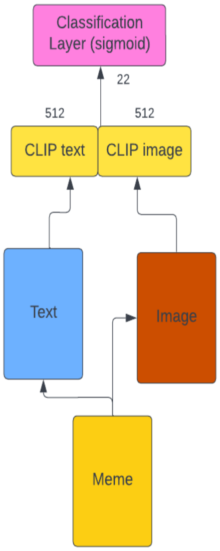

Figure 8. Using only CLIP features

### 4.1.2 Using CLIP features and BERT

We concatenated text and image features from CLIP and passed them through to a simple MLP. However, in memes, it is possible that text and image have no similarity. Thus, we are loosing information by not passing sentence separately. We added sentence embeddings from distilled version of BERT sentence transformer. Further, to prevent overfitting, dropout was added along with batch normalization. We tried another configuration of the model by fine tuning the last layer, unfreezed, of distilled BERT for our datset. Thus, we have three features at this stage:

- CLIP text and image features
- distilBERT text embeddings
- aggreagated face embedding vector

Rather than combining all features, we pass them to another Fully connected layer and then concatenate the outputs.

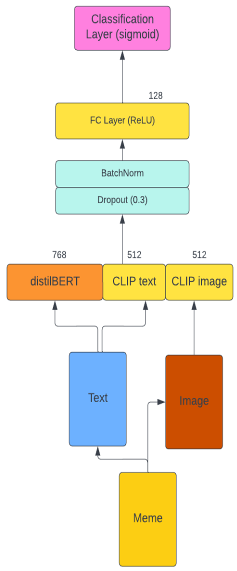

Figure 9. Using CLIP features and Bert features

### 4.1.3 Using CLIP features, BERT and facial features

We have observed that propaganda in memes have mostly targeted someone. To add this information, we find all faces in a meme using mtcnn from $facenet\_pytorch$. After that, we find embeddings for each face using $resnet$. We then do a mean pooling to get an aggregated vector representation for all faces.

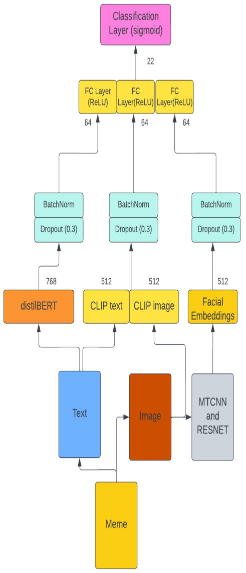

Figure 10. Using CLIP features, BERT and facial features

# 5. Experiments

## 5.1. Baseline

- Training loss decreased over epochs but there was no decrease in validation loss.
- There is visible over-fitting.
- We can see that losses are oscillating, hinting towards large weight values.

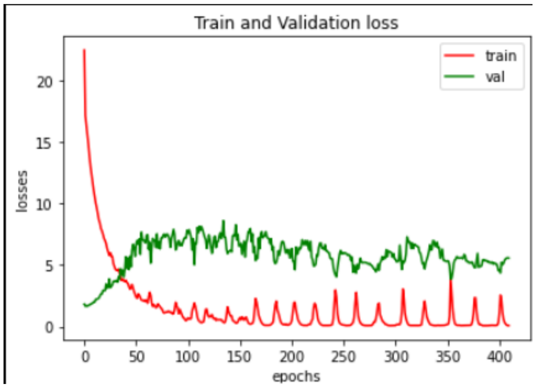

Figure 11. Training without transfer learning. Baseline model

## 5.2. Clip Basic

- The configuration was ran for 12 epochs.
- L2 regularisation was applied with $\beta = e^{-3}$.
- Adam optimizer was used with a learning rate of $5e^{-4}$
- Validation loss is always increasing and training loss increases after 2 epochs.
- There is no learning by the model.

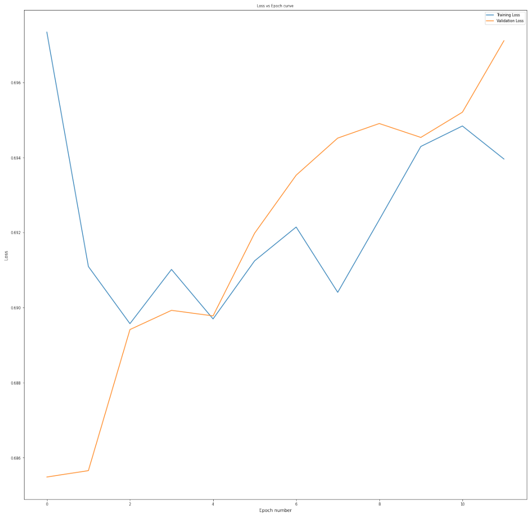

Figure 12. Loss vs epoch for Basic Clip model

## 5.3. Clip with BERT

- The configuration was ran for 12 epochs.
- L2 regularisation was applied with $\beta = e^{-3}$.
- Adam optimizer was used with a learning rate of $5e^{-4}$
- Both validation loss and training loss decrease over epochs.
- There is no visible over-fitting or under-fitting in the model.

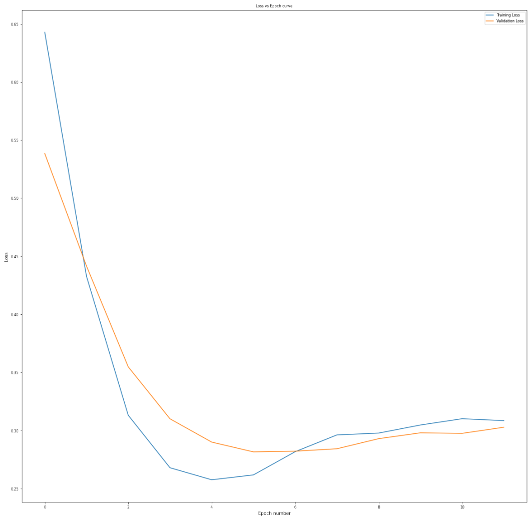

Figure 13. Loss vs epoch for Clip with distilBERT model

## 5.4. Clip with BERT fine tuning

- The configuration was ran for 12 epochs.
- L2 regularisation was applied with $\beta = e^{-3}$.
- Adam optimizer was used with a learning rate of $5e^{-4}$
- Both validation loss and training loss decrease over epochs.
- The validation loss is oscillating with training loss more than validation loss. There is visible under-fitting in the model.

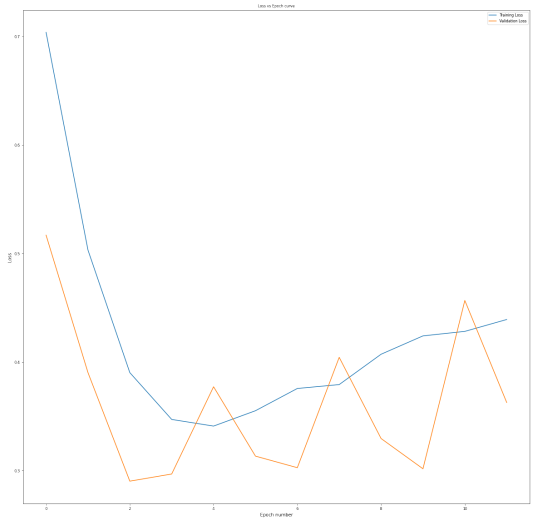

Figure 14. Loss vs epoch for Clip with distilBERT model with fine tuning

## 5.5. Clip, BERT and face embeddings

- The configuration was ran for 12 epochs.
- L2 regularisation was applied with $\beta = e^{-3}$.
- Adam optimizer was used with a learning rate of $5e^{-4}$
- Both validation loss and training loss decrease over epochs.
- There is no visible over-fitting or under-fitting in the model.

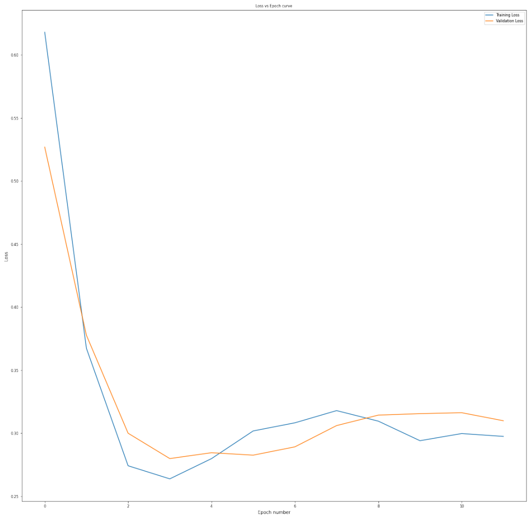

Figure 15. Loss vs epoch for Clip, distilBERT and face embeddings

| Model | f1-micro | f1-macro |
|---|---|---|
| VisualBERT + Coco [3] | 48.34 | 11.87 |
| Baseline Model | 43.6 | 15.6 |
| CLIP only | 32.57 | 4.19 |
| CLIP + BERT | 67.57 | 14.64 |
| CLIP + BERT with fine tuning | 69.82 | 17.13 |
| CLIP + BERT + face embeddings | 65.34 | 16.38 |

Table 1. Results from experiments. f1-score for each model

## 6. Conclusion

The task of multi-label multi-class classification was limited by the small size of dataset. However, using transfer-learning techniques, we were able to get results higher than the baselines mentioned in [3].

## References

[1] David M. Beskow, Sumeet Kumar, and Kathleen M. Carley. The evolution of political memes: Detecting and characterizing internet memes with multi-modal deep learning. *Information Processing  Management*, 57(2):102170, 2020. 3

[2] Yuyang Chen and Feng Pan. Multimodal detection of hateful messages using visual-linguistic pre-trained deep learning models, 2022. 3

[3] Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. Detecting propaganda techniques in memes. 2021. 1, 2, 3, 9

[4] Thilagavathi G, Kamesh M, and Boobala Vignesh P. *Image and Textual Sentiment Analysis of Memes, International Journal of All Research Education and Scientific Methods (IJARESM)*. 2021. 3

[5] Aijing Gao, Bingjun Wang, Jiaqi Yin, and Yating Tian. Hateful memes challenge: An enhanced multimodal framework, 2021. 3

[6] Kapil Gupta and Cheshta Kwatra. Multimodal meme sentiment analysis with image inpainting. In *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, pages 1–6, 2021. 2, 3

[7] 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. *Advances in Neural Information Processing Systems*, 33:2611–2624, 2020. 3

[8] 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. 3

[9] Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. *ECCV 2020*, 2020. 3

[10] Shraman Pramanick, Dimitar Dimitrov, Rituparna Mukherjee, Shivam Sharma, Md. Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. Detecting harmful memes and their targets. 2021. 3

[11] Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Md Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. Momenta: A multimodal framework for detecting harmful memes and their targets, 2021. 3, 4

[12] 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, 2021. 3, 4

[13] Riza Velioglu and Jewgeni Rose. Detecting hate speech in memes using multimodal deep learning approaches: Prize-winning solution to hateful memes challenge, 2020. 3

[14] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Making visual representations matter in vision-language models. *CVPR 2021*, 2021. 3