Report- Debangan Mishra (2022101027)

Task 1

The first task involves object detection within the memes followed by analyzing the distribution of the objects among the memes. The model used for this task was DETR-Resent-50 from HuggingFace. This model uses transformers for object detection in images and returns bounding boxes, class labels and confidence scores for their predictions. This model was chosen because of the following reasons:

1. The DETR model is trained on the COCO dataset which is expansive and has many class labels which ensure that a wide variety of objects can be detected within the memes.
2. Other models such as YOLOv8 too have been chosen, but the hypothesis (which was confirmed later) was that the first part of object detection within the memes will not provide significant results.

The model was let to run on the original memes. It was seen that the most predicted object was “person”, which is predictable as most memes tend to focus on people.

To get more detailed information about our model, I have included the deepface library which uses an ensemble of models for detecting information such as age, race, etc. from images. I have collected information about the emotion, age and race of the people as most memes tend to be offensive about race and gender, and emotion provides additional cues about the meme subjects. Other features that may have been detected were age, but was not included as a relatively low number of memes were ageist.

The deepface library provides various backend face detectors. According to their documentation, “*RetinaFace and MTCNN seem to overperform in detection and alignment stages but they are much slower. If the speed of your pipeline is more important, then you should use opencv or ssd. On the other hand, if you consider the accuracy, then you should use RetinaFace or MTCNN”.* I tested the default opencv face detection as well as RetinaFace detector. The results are very different with the RetinaFace detector being able to detect a lot more human faces and their races. The results are shown below:

Race detection libraries (deepface) do not seem to work that well, as there is a discrepancy in the number of people detected in the images and the number detected by the object detection model (detr-resnet-50)
There seem to be very few models that classify race and gender, thinking of these two as prime characteristics as most images tend to have persons, so cataloging only person should not work.

Deepface performance with the default detector: on an average every file has one less person detected than actual. This does not seem extreme, however, there are 16631 missing people as compared to the total people detected by detr-resnet-50 model. Note that even the DETR model does not always give correct counts for the humans detected. Occasionally there are errors. Yet the total number of faces detected is much less (23566 as opposed to 6935)

	person_count	person_count3	difference
count	9618.000000	9618.000000	9618.000000
mean	2.450198	0.721044	-1.729154
std	2.663293	1.055030	2.593604
min	1.000000	0.000000	-15.000000
25%	1.000000	0.000000	-2.000000
50%	1.000000	1.000000	-1.000000
75%	2.000000	1.000000	0.000000
max	22.000000	17.000000	13.000000

Deepface performance with the RetinaFace detector:

	person_count	person_count2	difference
count	9618.000000	9618.000000	9618.000000
mean	2.450198	2.329382	-0.120815
std	2.663293	4.539528	3.835600
min	1.000000	0.000000	-13.000000
25%	1.000000	1.000000	-1.000000
50%	1.000000	1.000000	0.000000
75%	2.000000	2.000000	0.000000
max	22.000000	99.000000	94.000000

We can see that a lot more people are being detected on average, and the difference magnitude is much smaller but took a long time to run (~9 hours as opposed to ~5 hours for the opencv detector to run). 23566 people were detected by DETR in total and 22404 by the RetinaFace model so the numbers are fairly close, but standard deviation is very high. In the following image, 99 people detected by RetinaFace are closer to the actual count as opposed to the 5 detected by DETR.

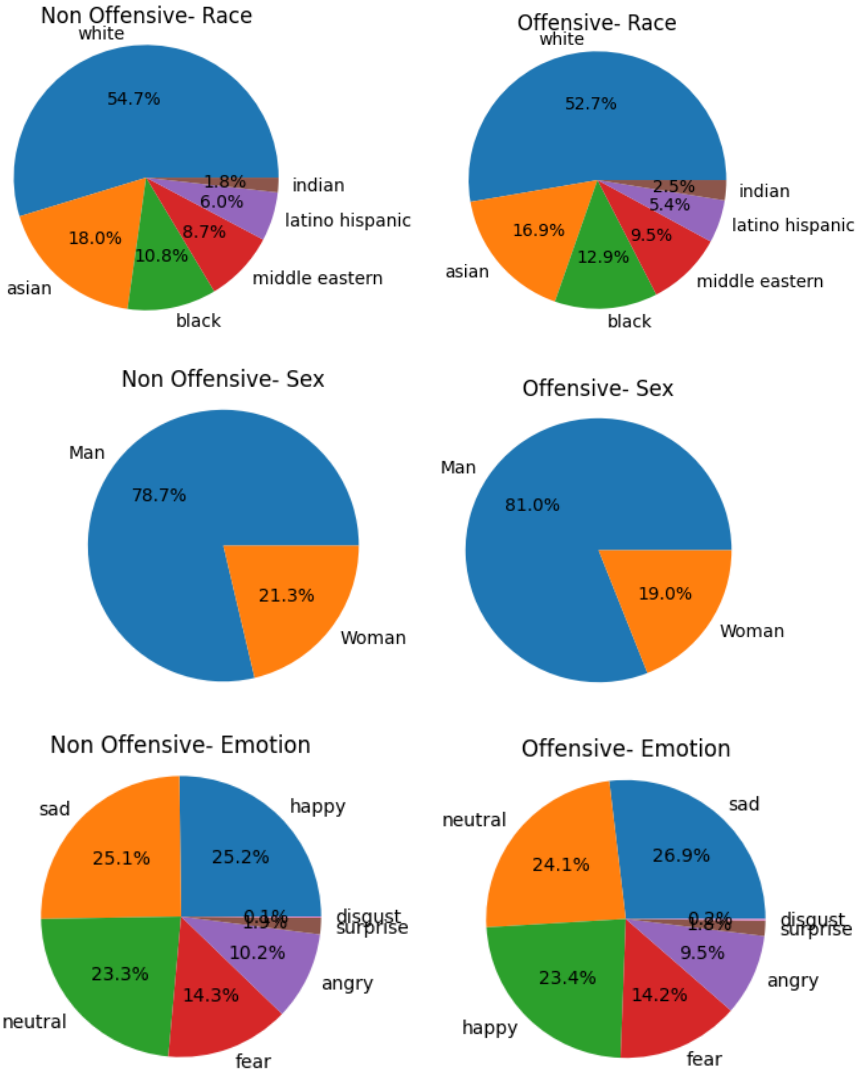
I have chosen the data of the RetinaFace model because it gave a closer result.
Procedure for this task involves running the DETR model in all the images, finding a list of files where people are detected and running the deepface VGG16 models in these files to detect- emotion, race, sex of the people depicted. Note that the detector model could have been run on all the images, but the object detector model is the main form of detector, and race detection is a supplement to that. I have not run this model on all the

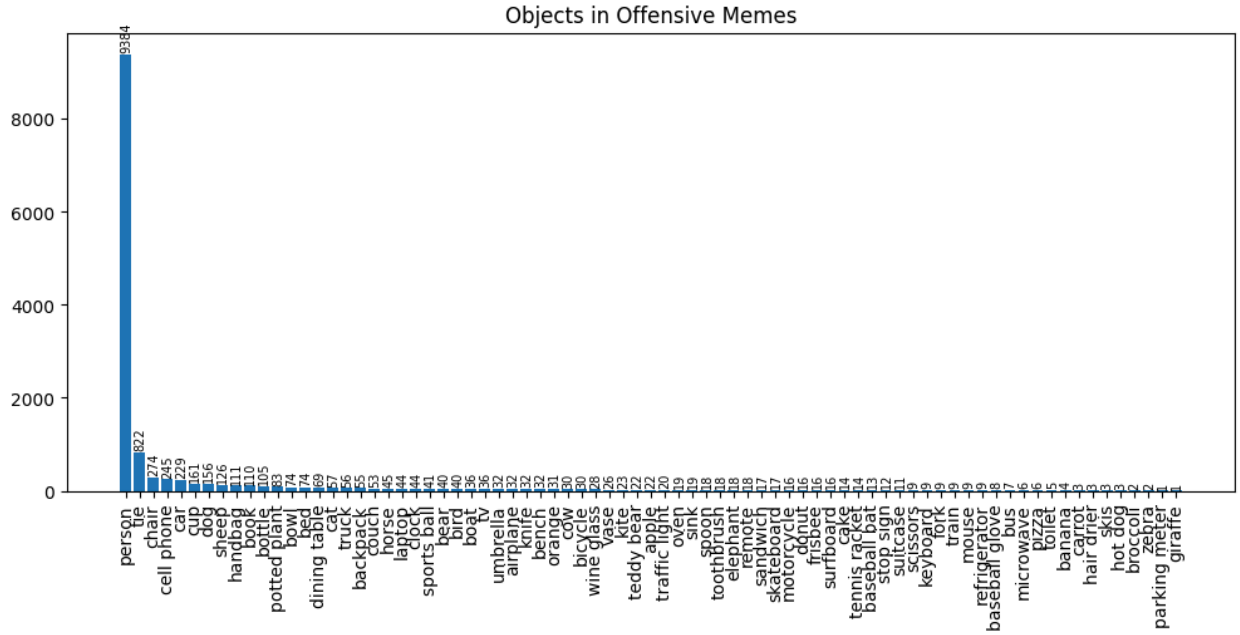
images in interest of time and also to prevent a scenario where no person was detected by my DETR model but deepface did and gave values of race, emotion and

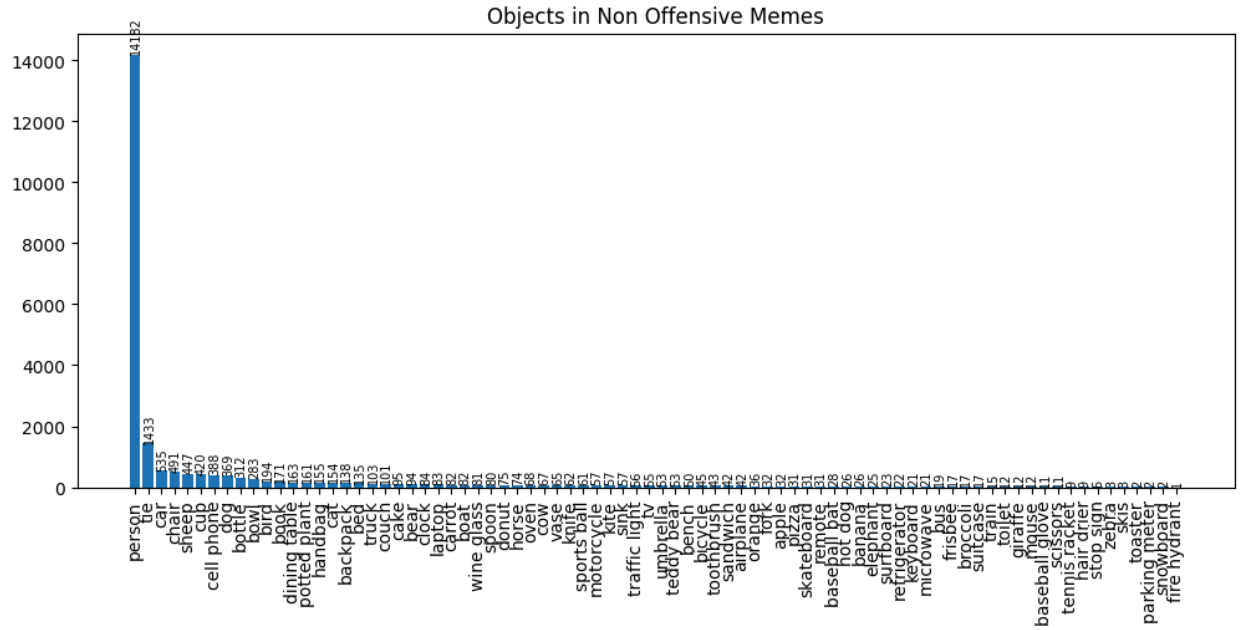
Potential problem discovered while doing this: images are duplicates and/or very similar. Same image is being used multiple times in the dataset: 05296.png, 46197.png, 06931.png, 19386.png with slightly different captions. Additionally some memes have different labels (hateful or not), different images altogether, but the text is the same. This reinforces the multimodal nature of the problem.

The results of the models are then analyzed and their distributions plotted. Unsurprisingly, the distribution of the objects tends to be nearly identical for both the offensive and non-offensive memes. Furthermore, all images have an extraordinarily high proportion of humans (this was the reason for choosing an add-on model for further classifying the humans detected by the model).

When we however try to see the category distribution of humans within the images, they too tend to have very similar proportions. The following graphs are seen







Although in total, the images might have intra-image distributions, such as a person occurring together with a dog, etc. Using this as an inspiration, classifiers were developed on a catalog- where the number of objects detected within each image is clubbed and a classifier is trained on the basis of this. Note that the number of classes is expansive and to avoid the curse of dimensionality, I thought of using PCA but the variances are fairly evenly distributed. Hence, using the original categorisation present in the COCO dataset (over which DETR is trained), the objects like airplane, cycle etc. are clubbed as Vehicles, and so on. This reduces the number of features to 30. Following this, SVM (RBF kernel), Logistic Regression Model, and Random Forests were trained, however it was noticed that the F1 scores were extremely low. Once data imbalance was mitigated by SMOTE, undersampling and oversampling, there was a significant improvement. However the AUROC scores are fairly low around 0.50. We can clearly see that our hypothesis of the necessity of multimodality still holds. Off the scenes, even YOLOv8 was used but results were very similar.

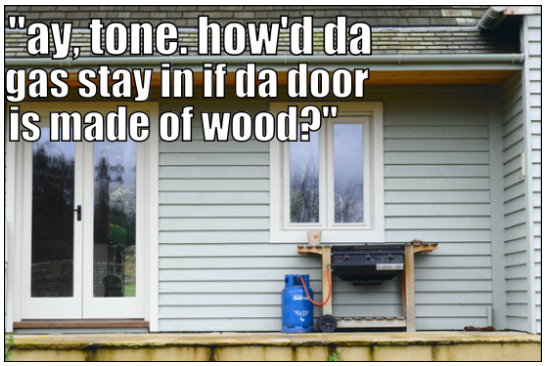
Task 2

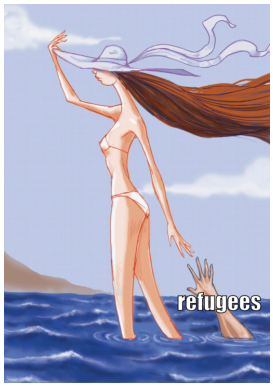
For this tak, a few new metrics were designed- Appearance, Disappearance, Confidence Increase, Confidence Decrease, IOU correspondence and label change. Idea was to process the images in three different techniques- fast marching technique (openCV's Telea inpainting method), the openCV inpainting method based on Navier Stokes, and a cropping method. For the first two cases, masks were prepared which were in-painted by the algorithm. Note that a threshold of 90% was kept for the object detection, which is rather high but it was seen that with above 90% confidence, the predictions are rather accurate. The method remains the same and thresholds can be adjusted according to one's needs. For cropping, the text bounding boxes were identified by easyocr, merged and then the largest non text region was identified in the vertical orientation, followed by cropping out the images in that region.

Note that the models used themselves are not always correct and can have inaccuracies. The following bounding boxes show that text has been detected, but that is incorrect:

Post the preprocessing steps on the images, the object detection algorithms were run on all the images. Now, the objects detected in the two images need to be mapped to each other, so that an effective comparison can be made. Hence, for this purpose, the IOU, (intersections over unions) were calculated between every pair of objects detected (Note that this preprocessing step is also time consuming and performing it for more images detected by a lower threshold would also be computationally ineffective). The IOU represents what fraction of the bounding boxes match, and a threshold of 50% was set, indicating that if a box matches less than 50% with any other box, it shall be counted as a box that has disappeared/appeared. The main base for the evaluation is the change in the confidence scores. Among the objects that are retained, if their labels are the same the confidence increase and decrease are mapped separately. In case their label changes, the confidences are added up. The appearance and disappearance indices are also similarly summed up.

There are some images where absolutely no objects are detected in any of the methods, primarily due to shortcomings of the model itself. (Smoke, Illustrations, etc. are not detected)





The results were that the methods elicited above do not work effectively and the model's object detection is fine as is. Hypothesis was a high appearance score for crop as cropping might lead to removal of objects from this image, and this hypothesis was confirmed. The disappearance factor was also high indicating that some objects detected in cropped images were no longer detected in the original images.

Some Interesting Qualitative Observations

In the following image, a "chair" object was detected only upon cropping and it was not detected in any other methods.

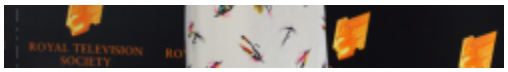
Similarly, only the cropped version detects a handbag in the following image, but nothing is detected in any other method. Although this is a garbage bag initially, handbag is still fairly close.

Additionally, only cropping detected a dog in the following image. Such qualitative analysis justifies a high appearance score for cropping but that poses inaccuracies.

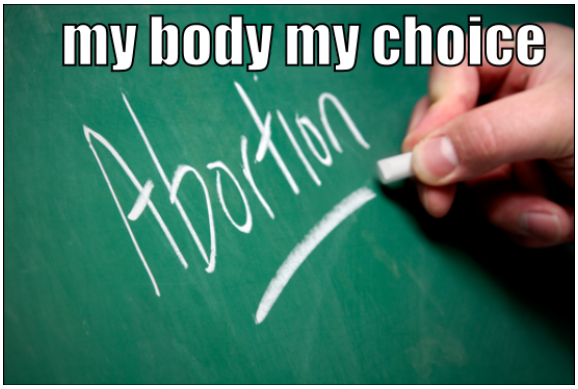
The adjoining image to the left is after processing by NS, and the one to the right is after processing by telea, and although both the modified images look similar, the model was able to detect humans in only the one processed by NS.

It is however also seen that the iou_arr parameter is quite low in cropping so we can conclude there is not a good fit in the bounding boxes of the images, as the bounding boxes tend to shift significantly when the images are cropped. With inpainting, no objects were any longer detected in some files. This happened with 80 files in NS and crop, and with 40 files in Telea.

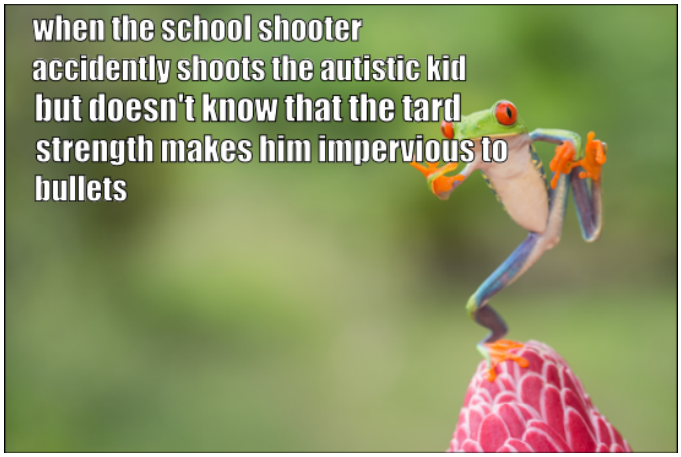
Cropped images often mean that background text which may or may not be considered useful can be detected and cause incorrect modifications or may be incorrectly blurred out. This is an example of that:



This particular meme will become useless post preprocessing due to it being majorly text:



In this case, regular preprocessing would mean that the object is covered up causing inaccuracies.



Task 3

This task involved creating an image classifier. I created a classifier- whether the memes are hateful or not. I trained two models- an early fusion model and an intermediate fusion model inspired by the paper “CLIPper: Multimodal Hateful Meme Classification based on Cross-modal Interaction of CLIP Features”.

Part 1:
The first model trained was a concatenation of the image embeddings generated by ResNet152, and the text embeddings generated by BERT. It is to be noted that these models are pre trained and were not fine tuned due computational limitations. Additionally, these models just gave an embedding of the images and text. Furthermore, the BERT embeddings were not for the word, but rather for the sentences as a whole, because although BERT can give individual tokens for the words, it also pools them and provides an embedding for the sentence as a whole. These two were concatenated and passed to two trainable networks - an RNN, and another MLP network. The metrics were better for the model RNN model, and there was a decrease in the losses over the epochs and the model converged to a certain degree. Yet the precision was still very low (possibly due to dataset imbalance).

Part 2:
This model involves generating CLIP embeddings of the images and the text. CLIP by OpenAI is a multimodal model that is able to generate embeddings of both texts and images in the same space. This is enabled by the training method in which similar text and image embeddings are penalized for being too far. However, this does not suit our purpose alone, as the images and texts can most definitely be different when the intention is sarcasm or offensiveness. Hence, the embeddings generated by the CLIP text and CLIP image models are passed through their own individual networks, after which corresponding elements of the two representations are multiplied and passed through a third network which is responsible for the classification of the images. It was observed that training all three networks with separate optimisers (Adam for the encoders and SGD for the outputter or Adam for all three) performed well with all the metrics being sufficiently high, and better than all other models (only image, only text, text+image concatenation).

Additionally, various configurations were tested (changing hyperparameters, activation functions,), including taking the entire flattened product matrix. The observation was that the model performed well in this case too, but the training process was slower. During training, the average accuracy was good enough, however, there was a drop in the AUROC score on the validation and test datasets, maybe due to noise introduced by the high dimensional feature space. An interesting observation was that the model performed badly when SGD optimisation was used for the image and text encoders. Usage of Adam optimization (without a learning rate decay) gave good results for all metrics as well, and Adam-Adam-SGD performance was comparable to Adam-Adam-Adam.

Task 4

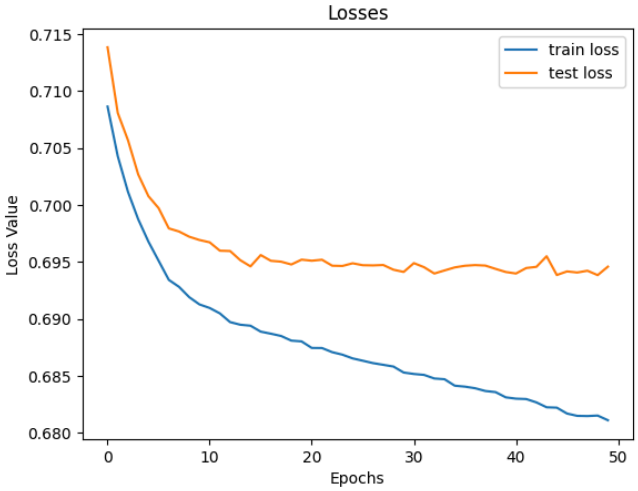
The task was to check if we can use only the text to make predictions about the offensiveness of the meme. Similar to Task 1, the hypothesis was that using only one modality will not be sufficient and will not lead to proper results. The preliminary data analysis involved preprocessing the captions to remove the stopwords, unnecessary information and punctuation. This was followed by Count Vectorization to get a numeric representation. Additional features such as sentiment, objectivity and hate word count of the text were introduced. The models trained were Complement Naive Bayes (a version of naive bayes that is used to deal with imbalance in data), Multinomial Naive Bayes, and Logistic Regression, because of the popularity of these models in establishing baselines in NLP tasks. It was observed (similar to the Task 1) that model F1 scores do not tend to be good enough in unbalanced sets, hence oversampling and undersampling were done (Not

SMOTE because it is not very accurate when dealing with NLP tasks). However, Complement Naive Bayes performed decently and similarly in all three cases (thereby attesting to its use in unbalanced datasets). There was an improvement in the metrics of the other two models as well, but still not close enough to the baseline models. Note that the models here tend to perform better than the ones in Task 1 as the text gives more verbose information about the meme.

In order to bring in some information about the image as well, the BLIP caption generator model was used. There is an argument in the BLIP model which is a text precursor. Adding the precursor “a photo without text” helped in the results being devoid of noise that could come from the captions. For example look at row 1 of the following dataframe.

Non-Conditional Image Captioning	Conditional Image Captioning
there is a man with a turban on his head and a quote on the side	a photo without text of a man with a turban on his head
there are two men that are standing in front of a herd of sheep	a photo without text of a group of men in white turbans with sheep
there is a dog that is standing in the grass with its tongue out	a photo without text of a dog with a caption of a human says who's a
there is a young boy that is holding a stuffed animal	a photo without text of a boy in a military uniform with a stuffed animal
a black and white photo of a man with a mustache	a photo without text of a man in uniform with a mustache
there is a dog that is looking at the camera with a caption	a photo without text of a dog with a caption of a caption
there is a woman that is smoking a cigarette and has a caption	a photo without text of a woman smoking a cigarette
there is a squirrel that is sitting on a tree trunk	a photo without text of a squirrel sitting on a tree trunk
there is a man and woman hugging on a bed together	a photo without text of a man and woman hugging on a bed
araffe sitting on a white cube with a caption saying kermi the frog definitely not	a photo without text of a frog sitting on a box
there is a man wearing a hat and jacket sitting on a couch	a photo without text of a man wearing a hat and jacket
there is a man in a suit speaking into a microphone	a photo without text of a man in a suit speaking into microphones
there is a young boy smiling and holding a toothbrush	a photo without text of a child smiling and holding a toothbrush
a picture of a picture of a man hanging on a rope	a photo without text of a picture of a man hanging on a rope
there is a man with a fake face eating a piece of food	a photo without text of a man eating a piece of food
there is a man in a space suit with a helmet on	a photo without text of a man in a space suit with a helmet on
someone is holding a black liquid in their hand with a dripping liquid	a photo without text of a hand with a dripping liquid
arafed image of a man in a red robe sitting in a chair	a photo without text of a man in a red robe sitting in a chair
arafed image of a group of men standing in front of a building	a photo without text of a group of men standing in front of a building
arafed man with purple hair and a purple scarf with words on it	a photo without text of a man with purple hair and a purple scarf

Following this, text embeddings were generated for both the caption and the description, and the embeddings were concatenated and used to train an RNN model. During the course of the training, it was seen that the accuracy improved and the losses decreased. However, the BERT embeddings were generated on only a small subset of the dataset, because of computational limitations. This however shows potential for future analysis - concatenating/ early fusion of BERT embeddings of caption and description. Due to training on a very small dataset, the metrics were not satisfactory though.



Scope for Improvement:
Some memes are offensive because of context and not because of either the image or the content, for example a meme about pigs might be considered offensive during muslim festivals in particular even if the comment is as simple as “Enjoy Pork” and similarly, a meme about steaks with caption “beef is delicious” might be considered offensive on a hindu festival. By definition: “A direct or indirect attack on people based on characteristics, including ethnicity, race, nationality, immigration status, religion, caste, sex, gender identity, sexual orientation, and disability or disease. We define attack as violent or dehumanizing (comparing people to non-human things, e.g. animals) speech, statements of inferiority, and calls for exclusion or segregation. Mocking hate crime is also considered hate speech.” from paper: “The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes”.

It is very subjective and depends on context, and it is difficult to expect machine learning models to know about such contexts like history, present events, cultural specifications and all. Additionally, this task is also very specific to countries and cultures. In India, Bengalis are often taunted as being able to do black magic, so the following meme can be considered offensive.

One possible solution: having a system- main model which extracts relevant features from the text and images- separately and also in a multimodal manner, and run it through the model, if confidence of predictions is not sufficient then a search engine query is made, and the context from the response is analyzed. A smaller version of this- incorporation of additional information to provide the models with context clues - is present in most of the top models of the Hateful Memes Challenge.

For object detection improvement, we can use Generative Fill

We can see that arguably, the object detection may improve and the object bounding boxes expand to include more information. However, this was not done extensively in my project primarily because of a lack of good open source generative fill images available online. There were various dependency clashes in trying to clone available repositories. Using Generative Fill would mean a significant impact to the current caption removal analysis method which relies on bounding boxes analysis. Yet, the same approach can be modified to see if the bounding boxes endpoints lie close to each other or not (within a margin of error), and if so, then finding their IOU, in order to map the objects to each other. Another improvement might be GNN on scene graphs as shown below to analyze properly what actions are taking place in the image.

Models and External Dependencies used:

1. Salesforce blip (Image Captioning)
2. Easyocr (OCR)
3. Retinaface on deepface (Face detection followed by Race/Gender/Emotion using VGGFace)
4. Detr-resnet-50 (Object Detection)
5. BERT (Embeddings)
6. CLIP (Embeddings)

Note: The github submissions do not include many of the pickle files that were used, nor does it include many of the processed images as it would massively blow up the storage required.