UTILISATION OF ARTIFICIAL INTELLIGENCE FOR MULTIMODAL UNDERSTANDING OF MEMES

Kaitlyn Janine Ang En[1], Kuek Yong Jie Adriel[2], Wong Minn Xuan[2]
[1]Raffles Girls School (Secondary), 2 Braddell Rise, Singapore 318871
[2]DSO National Laboratories, 12 Science Park Drive, Singapore 118225

Abstract

In the era of digital communication, memes have become a pervasive form of expression, conveying humor, satire, and cultural references. This computer vision project explores the application of artificial intelligence (AI) to discern between memes and non-memes within diverse visual content. Leveraging state-of-the-art deep learning models, the system employs advanced image recognition techniques to analyze and classify images as either a meme or a non-meme, contributing to a deeper understanding of online visual culture.

The project involves training a convolutional neural network (CNN) on a carefully curated dataset of memes and non-memes, encompassing a wide spectrum of visual styles and contexts. Through iterative model refinement and optimization, the AI system learns intricate patterns, contextual nuances, and humor cues inherent in memes. The resulting model not only distinguishes between the two categories but also provides interpretable insights into the features contributing to its classification decisions.

The implications of this research extend beyond mere categorization, holding promise for content moderation, cultural analysis, and the development of more nuanced AI-driven communication tools. By enhancing our ability to automatically identify and understand memes, this project contributes to the ongoing evolution of AI technologies that navigate the rich landscape of online visual content.

1 Introduction

Memes are a common form of communication in the digital age. However, with popularity comes the rise of internet users taking advantage of this medium in order to create something to deliberately spread hate towards an individual or community. Thus such ideas should be stopped before they can cause too much damage, giving rise to developing models that can pick apart memes from non-memes in order to more easily find hateful memes.[1] However the task of classifying memes and non-memes is a challenging one, as memes can take many

forms and often rely on cultural references and humor. But recent advances in deep learning have made it possible to train models that can accurately classify memes with high precision and recall.

An example of AI being used for memes, while not specifically classifying memes and non-memes, is the Facebook Hateful Memes Challenge[3][4]. Participants across the world were tasked to develop multimodal machine learning models in order to identify whether a meme was hateful or not. Through combining text and image feature information, participants were able perform well, with the top 5 teams having AUC ROC scores of 0.79-0.84 on the unseen data; higher than the baseline provided in the original paper.[4]

In this report, I describe my efforts to train an AI model to classify memes and non-memes. I will begin by discussing the dataset I used for training and evaluation, and then describe the architecture of my model. Finally, I present the results of our experiments and discuss the implications of my findings.

2. Materials and Methods

Due to the multimodal nature of most memes, it was crucial that the model would be able to well understand both the text and image and make correlations between both in order to determine whether a given image is a meme or a non-meme. While there are no pre-existing models or papers that specifically tackle the topic of identifying memes, much of the basis of this project was based on the first place entry for the Facebook Hateful Memes Challenge[4].

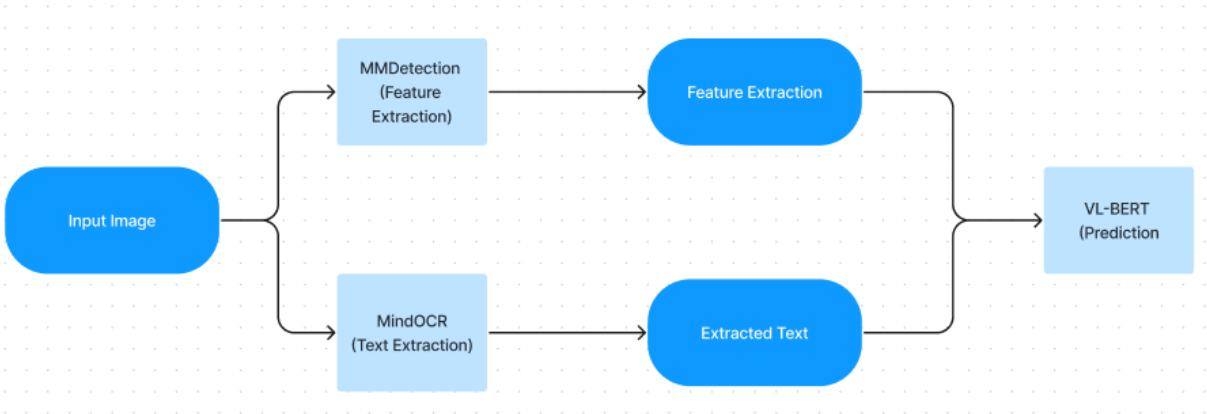

Fig 1. The rough pipeline of the model

Stage	Model Used	Usage
Stage 1	MindOCR[7]	Text Extraction

	MMDetection[8]	Feature Extraction
Stage 2	VL-BERT[9]	Final Prediction

Table 1, models used for this task and their purpose

2.1 MindOCR

An optical character recognition (OCR) tool used for extracting text from images. MindOCR allows the model to be able to utilise the linguistic features of the images fed into it in order to have a clearer idea of whether the image is a meme when paired with the visual features. DB++[5] was used for text detection while CRNN[6] was used for text recognition.

2.2 MMDetection

For memes, the synergy between the textual and visual components is crucial. Feature extraction plays a crucial role in enabling the model to discern these connections between text and image, facilitating more accurate meme classification. By training MMDetection on diverse datasets with varied classes, it is able to enhance the model's ability to correlate images and text, ensuring robust predictions for meme identification.

2.3 VL-BERT

Between the e2e and prec pretrained models for VL-BERT, e2e was eventually chosen for its ability to make semantic alignments between image and text and learning of visual representation[10] - a crucial component in learning how to differentiate between memes and non-memes

2.4 Datasets

For MindOCR, the Total Text Dataset[11] was used. It consists of 1,555 images with a variety of text types including horizontal, multi-oriented, and curved text instances. There are 1255 training and 300 testing images.

For MMDetection, it was trained on the combined datasets of PascalVOC 2007[12] and 2012[13], which contains 20 different object categories, 1464 training images, 1449 validation images and a private testing set

For VL-BERT, it was to be fine tuned on VQA[15][16][17], RefCOCO+[18]. The VQA dataset contains open-ended questions about images, with each image having at least 3 questions, 10 ground truth answers per question and 3 plausible, but likely incorrect answers per question, allowing for the model to have a better understanding of vision, language and commonsense knowledge; while RefCOCO+ consists of 141,564 refer expressions for 49,856 objects in 19,992 images. Afterwards, the model was to go through further fine tuning on a dataset which consisted of memes taken from public domains on the internet and social media platforms, adhering to ethical considerations and respecting copyright regulations. This last dataset contained exactly 500 instances of both memes and non-memes, which were later resized to 512x512 sizing for easier training. The dataset represents as many scenarios as possible, covering the different types of memes and what topics they can be about.

3. Results

3.1 Stage 1 Results

After training, both MindOCR and MMDetection were able to perform their tasks with relative accuracy; able to produce relatively accurate results for their respective tasks (Fig 7.1,7.2 and 8 in Annex). Further discussion of the limitations and challenges encountered during the training of these two processes will be provided in the Discussion section.

3.2 Stage 2 Results (Incomplete)

Despite efforts to train VL-BERT on the self-curated dataset, the process remained incomplete, resulting in the model being unable to generate predictions. Technical challenges, including data preprocessing issues and compatibility constraints with the model architecture, hindered the completion of the training process. As a result, the performance and effectiveness of VL-BERT in meme classification could not be assessed in this project.

4. Discussion

While the model is still able to predict the category of a given image, there are still many aspects in which this model can be improved on.

4.1 Improvements to pre-existing models

Firstly, due to time and computational constraints, both MindOCR and MMDetection were only trained once.

Further training of MindOCR on the SCUT-CTW1500[19] dataset, which contains 1,500 images: 1,000 for training and 500 for testing. In particular, it provides 10,751 cropped text instance images, including 3,530 with curved text. The dataset contains a lot of horizontal and multi-oriented text. It should also help in preventing the text extractor from unnecessarily lumping words together into one longer text (Fig. 2.1 and 2.2 in Annex). There is also a noticeable challenge when it comes to certain types of fonts and smaller font sizes (Fig.3 in Annex), an example of which is when "I" is a rectangle, thus causing the model to be unable to detect the word correctly. (Fig.1 in Annex) This suggests the need to find more strategies in order to improve model robustness in tackling diverse text formats that are found in memes.

MMDetection could be further trained using the Visual Genome dataset[20], which consists of 101,174 images from MSCOCO[21] with 1.7 million QA pairs, 17 questions per image on average. Compared to the Visual Question Answering dataset. The Visual Genome dataset also presents 108K images with densely annotated objects, attributes and relationships. This would provide the model with more classes for detection, preventing wrong classification of objects found in memes(Fig. 4 in Annex). In addition, PyBottomUpAttention[22], also trained on Visual Genome, can be used as a further supplement for the feature extraction.

In terms of the data collection, while there are datasets and models that are able to create memes, I was not able to find any that consisted of both memes and non memes. Some types of memes require prior context of a certain piece of media in order to fully grasp the meaning of the meme, which would make it hard for VL-BERT to recognise that it is a meme and thus may result in a wrong categorisation. This also applies to images that contain subtitles of what the character is saying, where the format is similar to some types of memes, thus also resulting in the possibility of the model wrongly classifying it as a meme. Some memes are also done through animated characters, which MMDetection is not trained on and thus may not be able to recognise them correctly (Fig. 5.1 and 5.2 in Annex). This could also cause there to be multiple overlapping bounding boxes with different labels (Fig. 6 in Annex), causing ambiguity in the visual representation of the scene, resulting in a less accurate prediction.

In addition, text overlays may cover up important visual features of the image, thus requiring a way to get rid of them while still maintaining the features of the original image. Thus Inpainting models such as MAT[23] or MMEditing[24], trained on the CelebA[25] and Places365[26] datasets can be used to inpaint over text in such memes() to allow for a more accurate detection from MMDetection.

4.2 VL-BERT

As mentioned in section 3.2, I was unable to finish training my VL-BERT model. However, if I were to embark on this project again, I would also train it on the Memotion[2] dataset, which is a dataset specifically curated for the sentiment classification of memes, thus allowing the model to have a deeper understanding of the variety of memes.

Acknowledgements

I would firstly like to acknowledge my mentors, Adriel and Minn Xuan, for their invaluable guidance, support and mentorship throughout the duration of this research project. Their expertise and encouragement were instrumental in shaping the direction of my research and overcoming the various challenges I faced.

I am also thankful to DSO for providing access to resources, facilities, and funding that facilitated the execution of this project.

I would also like to acknowledge the authors and contributors of the datasets, tools, and resources used in this project. Their efforts in curating and sharing valuable data and software were essential to the success of this research project.

References

[1] Sharma, V. (2020, November 15). Introduction to Meme Classification using PyTorch and fastText. DataDrivenInvestor. Retrieved from https://medium.datadriveninvestor.com/introduction-to-meme-classification-using-pytorch-and-fasttext-e667a81b5e0

[2] Sharma, C., Paka, S. W., Bhageria, D., Das, A., Poria, S., Chakraborty, T., & Gambäck, B. (2020). Meme vs. Non-meme Classification using Visuo-linguistic Association. In Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020), Barcelona, Spain. Association for Computational Linguistics.

[3] DrivenData. (2021). Hateful Memes: Phase 2 [Webpage]. Retrieved from https://www.drivendata.org/competitions/64/hateful-memes/

[4] Kiela, D., Firooz, H., Mohan, A., Goswami, V., Singh, A., Ringshia, P., & Testuggine, D. (2021). The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes.

[5] Liao, M., Wan, Z., Yao, C., Chen, K., & Bai, X. (2019). "Real-time Scene Text Detection with Differentiable Binarization."

[6] Shi, B., Bai, X., & Yao, C. (2015). "An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition."

[7] MindSpore Team. 2023. MindSpore OCR :MindSpore OCR Toolbox

[8] Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., ... & Lin, D. (2019). MMDetection: Open MMLab Detection Toolbox and Benchmark.

[9] Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., ... & Dai, J. (2020). VL-BERT: Pre-Training of Generic Visual-Linguistic Representations. In: International Conference on Learning Representation.

[10] Xu, H., Yan, M., Li, C., Bi, B., Huang, S., Xiao, W., & Huang, F. (2021). "E2E-VLP: End-to-End Vision-Language Pre-training Enhanced by Visual Learning." arXiv preprint arXiv:2106.01804.

[11] Chng, C. K., & Chan, C. S. (2017). Total-Text: A Comprehensive Dataset for Scene Text Detection and Recognition.
[12] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2007). The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.
[13] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2012). The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.
[14] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The PASCAL Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, 88(2), 303–338.
[15] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. "Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering." In Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[16] Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. "Yin and Yang: Balancing and Answering Binary Visual Questions." In Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[17] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. "VQA: Visual Question Answering." In International Conference on Computer Vision (ICCV), 2015.
[18] Yu, L., Poirson, P., Yang, S., Berg, A. C., & Berg, S. L. (2016). Modelling Context in Referring Expressions.
[19] Chen, X., Jin, L., Zhu, Y., Luo, C., & Wang, T. (2020). Text Recognition in the Wild: A survey.
[20] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., ... Li, F.-F. (2016). Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. arXiv. Retrieved from https://arxiv.org/abs/1602.07332.
[21] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., ... & Dollar, P. (2015). Microsoft COCO: Common Objects in Context.
[22] Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2018). Bottom-Up and Top-Down Attention for Image Captioning and Visual Questioning Answering.
[23] Li, W., Lin, Z., Zhou, K., Qi, L., Wang, Y., & Jia, J. (2022). MAT: Mask-Aware Transformer for Large Hole Image Inpainting. arXiv. Retrieved from https://arxiv.org/abs/2203.15270.

[24] Chen, K., Wang, Q., Zhang, Z., Tang, M., Chen, J., Wang, K., Wang, T., Zhu, Z., Liu, Q., Lin, D., & others. (2022). MMEditing: OpenMMLab Image and Video Editing Toolbox. arXiv. Retrieved from https://arxiv.org/abs/2201.06028.
[25] Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep Learning Face Attributes in the Wild. In Proceedings of International Conference on Computer Vision (ICCV).
[26] López-Cifuentes, A., Escudero-Viñolo, M., Bescós, J., & García-Martín, Á. (2020). Semantic-aware scene recognition. Pattern Recognition, 102, 107256. doi:10.1016/j.patcog.2020.107256.

Annex

Fig. 1 (Above): An example of “I” being represented by a rectangle in a certain font

Fig. 2.1 (Above): Image provided Fig. 2.2 (Below): The results from MindOCR

```
kombsiooal      0.5474835634231567
single  0.9999997019767761
leavest 0.9845641255378723
and     0.9468973278999329
coffee  0.9944584965705872
black   0.9999998807907104
orders  0.9996313452720642
mcdonalds       0.9940860271453857
children        0.9999998211860657
as      0.999998927116394
through 0.9999633431434631
cheer   0.9998480081558228
drive   0.9661860466003418
the     1.0
into    0.9896031618118286
spulls  0.9965813159942627
wehaveloodathomen       0.9652403593063354
incdonadstncoronaris    0.8123834729194641
mcdonaldsi      0.9886615872383118
childrenyelling 0.9983850121498108
goopyheartifallyn       0.8261948227882385
```


```
kdst    0.8066580295562744
preelhiting     0.512956440448761
sexuaizing      0.8990877270698547
```

Fig. 3.1 (Left) and 3.2 (Right). 3.1 is in its original dimensions while 3.2 is the results given from MindOCR.

Fig. 4. An example of where the lack of a class for coins results in a wrong classification as a bottle, which would affect the understanding of the meme.

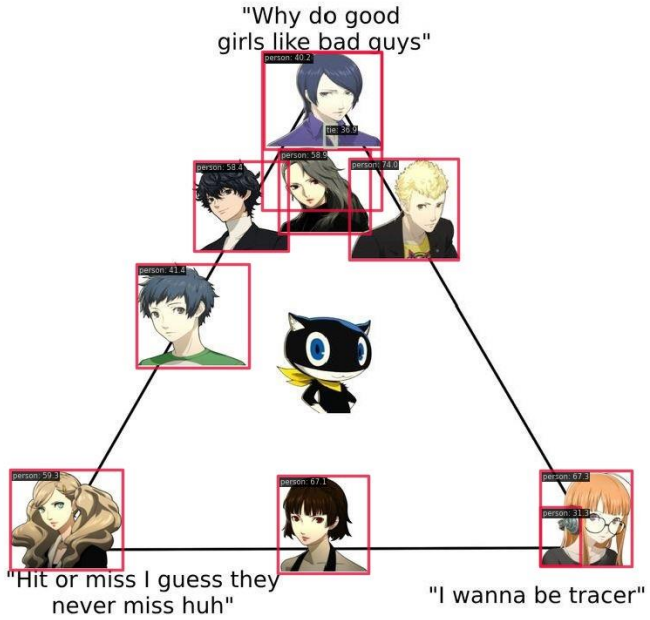

Fig 5.1 (Top) and 5.2 (Bottom). 5.1 Shows MMDetection being able to accurately detect the animated characters found in the image. 5.2: An example of MMDetection unable to detect and properly identify animated characters of another artstyle

Fig. 6. An example of multiple bounding boxes over the same location.

```
utensil 0.9824293851852417
kitchen 0.9895902276039124
favorite        1.0
my      1.0
one     0.9999998211860657
gotta   0.917290210723877
be      0.9999994039535522
of      1.0
are     0.9999804496765137
too     0.9999998211860657
plates  0.9742907881736755
bowls   0.9871147871017456
that    0.9232640266418457
shout   0.9960072636604309
out     0.9965067505836487
to      0.9999998807907104
```

Fig 7.1(Top) and 7.2(Bottom):MindOCR able to confidently recognise and identify words found in Image 7.2

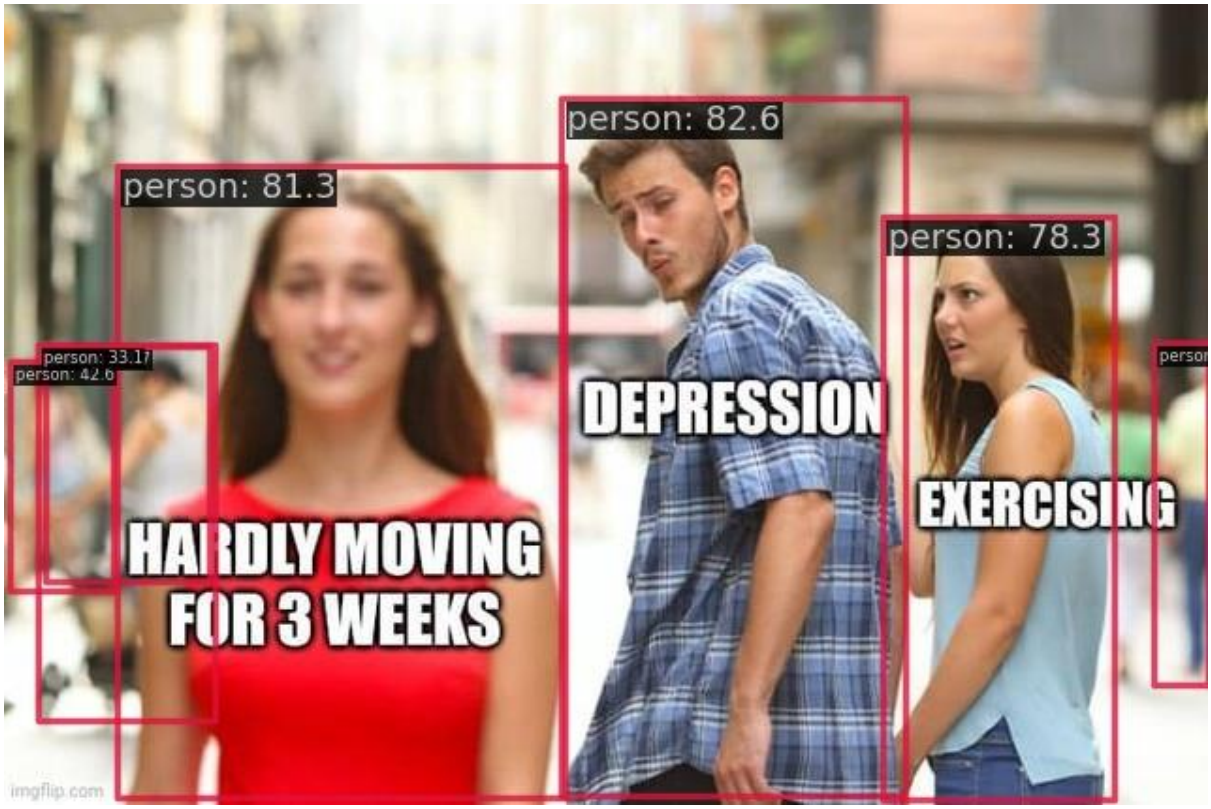

Fig. 8: An example of MMDetection able to accurately identify key visual features of an image