Project Report – Meme Template Classification

End Semester Assessment – CSE425

(5th November, 2024)

ARKA GHOSH
125018008

Table of Contents

Abstract

The future of marketing resides in memes. Leading organisations around the world have adopted memes as a form of advertising. However, unlike traditional methods of advertising, memes used for advertising must organically reach popularity like other memes shared on most social media platforms. Hence, determining which meme can potentially rise to popularity is imperative to any organisation looking to market through memes. The rise to popularity of memes, or virality, is a complex concept involving numerous variables. Therefore, the task of identifying such factors can be simplified using machine learning. This project aimed at using two different machine learning approaches to build a model that can identify meme templates, given the meme. The motivation was to tackle the first step in any form of research concerning memes, identifying the template a meme belongs to.

1. Introduction

A meme is an idea or concept shared primarily through imitation. The word's etymology stems from the Greek word, meaning ‘imitated thing’. The term ‘Internet Meme’ is used when an idea, behaviour, or style is shared through the Internet. Internet memes (hence forwards referred to as ‘memes’), originally evolved as a means of communicating humour through a shared medium. The motivation for creating a meme includes inside jokes within a certain community, recent events, social phenomena, etc. As memes started being used across the internet, they are now considered to be an integral part of ‘Internet Culture’. A wide range of research has been done on memes within the fields of marketing, finance, politics, religion, social movements, etc. Alongside memes, meme genres also evolved. Reflecting the dynamic nature of the internet, memes have a short lifecycle before new ones become popular. Meme genres although, relatively have a much larger lifecycle. Memes, and thereby extending to meme genres usually reflect the general trends of the real world. Since their primary purpose is to be a medium of shared ideas, their popularity is intertwined with the recency of their subjects.

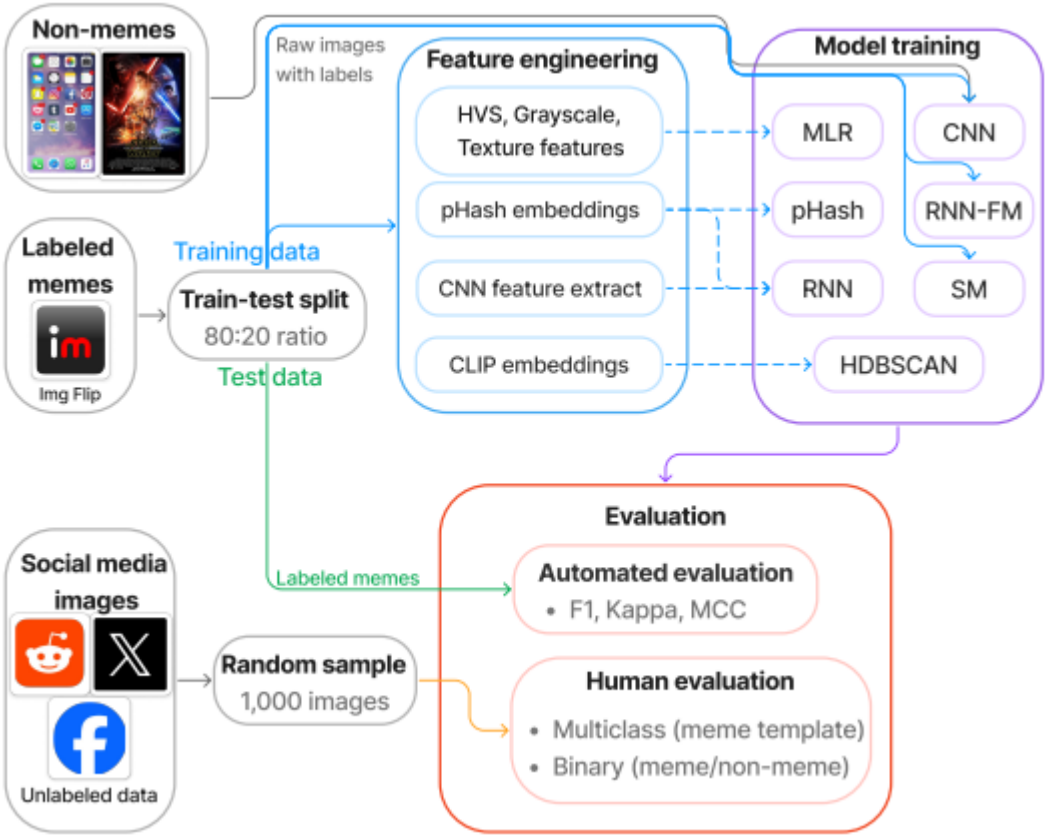
As the internet and social media platforms evolved, memes also evolved into being formatted into certain trending ‘formats’. These ‘formats’, otherwise known as ‘meme templates’, can be a picture containing specific visual elements with a blank space meant for people to add the ‘meme’. Templates usually change along with time and with the increased popularity of social media, newer templates get created faster than they did in the past. Meme template forms one of the factors in the success rate of memes on the internet, otherwise termed virality. Over the last few years, companies have shifted to using viral meme formats. This shift away from the traditional forms of marketing is motivated by the change in reception to traditional forms of marketing by the current generation. Outside of companies, other organisations such as political parties have also taken to creating accounts on the top social media platforms. Organisations have started hiring social media managers and influencers to generate internet content including memes and short-form content. This shift has put the spotlight on memes. It has now become crucial for companies to understand the factors required for a meme to become viral. Since individual ‘memes’ have an extremely short lifespan, more emphasis needs to be on the meme template. Going forward with subsequent research on meme templates can enable organisations to achieve better success at marketing through memes. This project takes inspiration from the above problem statement. Using the work done by Murgás *et al.* [1] as the base, Multinomial Logistic Regression and Convolutional Neural Networks are used to classify memes using their respective templates. This project mainly serves as a symbolic testament to the base paper.

The rest of the document is laid in 5 sections. Section 2 (Present Work) describes the work carried out in the base paper in brief. It covers the methodology and data collection steps. Section 3 (Background) goes over all the prerequisites necessary to understand the project at hand. Section 4 (Methodology) details the work done in the project. Section 5 (Results) displays the results of the work. Section 6 (Discussion) dives into some important outcomes of the project.

2. Present Work

2.1. Methodology

The proposed work in the base paper (Murgás *et al.* [1]) covers two groups of methods: supervised and unsupervised. To tune the dataset, the authors of the base paper employed several methods, baseline features (RGB histogram and greyscale histograms [2], and texture features via Local Binary Patterns [3]), embeddings via feature extraction (Derive embeddings by extracting features from the penultimate layer of the best-performing CNN model trained for meme template classification), Perceptual Hashing [4], CLIP (Contrastive Language-Image Pretraining, developed by OpenAI [5]), Oriented FAST and Rotated BRIEF (ORB) features (inspired by how humans perceive images). In model training, the paper follows the above-mentioned approaches. Under supervised learning, Multinomial Logistic Regression, Radius Nearest Neighbours, Transfer Learning and CNN, RNN Feature Matching, and Sparse Matching models were used. The goal here was to maximise Mathew's Correlation Coefficient. Under unsupervised, HDBSCAN [6], and Perceptual Hashing Method were used.



Proposed Methodology as per the base paper

2.2. Data Collection

The data for the project was collected from the following sources: Reddit, X (Twitter), and Facebook. Apart from these social media platforms, labelled data was collected from

ImageFlip. ImageFlip is a website that allows users to create memes using trending templates. The memes that are created are also available on the website for download. This paves the way for an annotated dataset of meme templates. To verify the authenticity of the models, certain non-meme images were also gathered. These included advertisements, screenshots, movie posters, and pictures. The data for non-meme images were randomly sampled from the Flickr30k dataset.

Source	**Type**	**Sample Instances**	**Template Instances**
ImgFlip	Memes	124,208	1,145
Reddit	Mixed	899,522	unlabelled
Facebook	Mixed	235,880	unlabelled
X (Twitter)	Mixed	174,338	unlabelled
Screenshots	Non-Memes	42,891	unlabelled
Advertisements	Non-Memes	41,462	unlabelled
Flickr30k	Non-Memes	31,783	unlabelled
Movie Posters	Non-Memes	8,052	unlabelled

3. Background

3.1. Choice of Models

Due to the vast nature of the base paper, the scope of this project was chosen to be limited only to 2 models out of the 7 demonstrated in the paper. The models chosen are as follows:

3.1.1. Multinomial Logistic Regression

Multinomial Logistic Regression (MLR) [7] is a more general version of simple logistic regression. Logistic regression involves classifying data into either a binary independent variable (dichotomous variable) or a continuous independent variable. This is made possible by using a logistic regression function that takes input in the form of independent variables. The output of the function, referred to as the dependent variable, is then collected and assumed to be the result of simple logistic regression.

In the case of MLR, the function used can classify the independent variables into multiple classes as required. The advantage of MLR is that it does not assume normality, linearity, or homoscedasticity. MLR uses a function known as a SoftMax function to process the independent variables. The SoftMax function for a class k in a multiclass setup is:

$$P(y = k|x) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$$

Where z_k is the linear combination of features for class k and K is the total number of classes.

MLR was chosen because it serves as a good model to justify as a baseline. According to the base paper, the model also gives sufficiently good results. The model was easy to implement since it used simple features (RGB, greyscale, and texture features).

3.1.2. Convolutional Neural Network

A Convolutional Neural Network (CNN) is a type of deep learning algorithm inspired by how human brains work. CNN has multiple fully connected layers. CNNs are best suited for tasks that involve visual recognition, such as image recognition. CNN has multiple layers some of the important layers are convolutional layers, pooling layers, activation functions, and fully connected layers. The CNN used in this project is a type of Densely Connected Convolutional Network, DenseNet121 [8]. The DenseNet model is a FeedForward model which has an advantage over traditional CNN models since it requires fewer parameters.

$$H_l = \mathcal{F}_\ell([H_0, H_1, \ldots, H_{l-1}])$$

Where,

- H_l is the output feature map at layer l,
- $F_l(\cdot)$ is the transformation function at layer l,
- $[H_0,H_1,\ldots,H_{l-1}]$ denotes the concatenation of all previous feature maps up to layer l.

CNN using DenseNet121 was chosen since deep learning gives the best result whenever there is a task of image recognition, like the current problem statement involving memes. This model also gave the highest results as per the base paper.

3.2. Data

The dataset required for the project was available in the public GitHub repository that was referenced in the paper. The images of the memes, however, had to be accessed via a request made to the authors of the paper. The request led to being granted access to a HuggingFace repository which contained 124,201 memes. The memes were labelled in their respective folders.

3.3. Preprocessing

The first step in preprocessing was to load the images using the path to the images provided in the dataset. In doing so, it was discovered that certain images were missing from the repository

cloned from HuggingFace. After filtering the dataset off images that were missing, the resultant sample size was 115,343.

3.3.1. Multinomial Logistic Regression – Feature Extraction

For training the MLR model simple baseline features were extracted from each image. To achieve this a pipeline function was created which extracted three types of features, HSV (Hue, Saturation, Value) features, greyscale features, and LBP (Local Binary Pattern) features.

3.3.2. Convolutional Neural Network – Transformations

Once the dataset was loaded and split into training and validation sets, several transformations were applied to the images based on which set they belonged. For the training set the extra transformations applied were Random Rotation, Colour Jitter, and Gaussian Blur. For both sets, summations were applied to convert to a tensor and to normalise.

3.4. Performance Metrics

The performance metrics used for all the models in this project are:

3.4.1. Mathew's Correlation Coefficient

Matthew's Correlation Coefficient (MCC) score is a metric used to evaluate machine learning models. It takes into consideration true and false positives and negatives. One advantage of the MCC score is that it can be used for different-sized classes.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Where:

- TP = True Positives – The number of positive samples correctly predicted as positive.
- TN = True Negatives – The number of negative samples correctly predicted as negative.
- FP = False Positives – The number of negative samples incorrectly predicted as positive.
- FN = False Negatives – The number of positive samples incorrectly predicted as negative.

3.4.2. Cohen Kappa Score

The Cohen Kappa Score is used to assess the agreement between different raters on an item belonging to the same class. It accounts for a classification being done by pure chance. Hence, it is a robust measure.

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

$$p_o = \frac{TP + TN}{TP + TN + FP + FN}$$

$$p_e = \sum_{i=1}^{C} \frac{(TP_i + FP_i)(TP_i + FN_i)}{N^2}$$

Where:

- p_o = observed agreement (the proportion of times the raters agree).
- p_e = expected agreement (the proportion of times the raters would be expected to agree by chance).
- *TP* = True Positives
- *TN* = True Negatives
- *FP* = False Positives
- *FN* = False Negatives
- TP_i is the number of instances that were classified as class *i* by rater 1.
- FP_i is the number of instances that were classified as class *i* by rater 2.
- *N* is the total number of instances.

3.4.3. F1 Score

The F1 Score is a harmonic mean of precision and recall. It helps provide a balance between both the stores especially if the class distribution is imbalanced.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

- *TP* = True Positives: the number of positive samples correctly predicted as positive.
- *FP* = False Positives: the number of negative samples incorrectly predicted as positive.
- *FN* = False Negatives: the number of positive samples incorrectly predicted as negative.

3.4.4. Confusion Matrix

A confusion matrix Provides a comparison between the actual classes and predicted classes.

4. Methodology

As discussed above, two different models were coded for this project, both under supervised learning. All files can be found at [9].

4.1. Multinomial Logistic Regression

This model was selected to provide a simple baseline. The model uses simple features that were extracted from each image. To build the model, exploratory data analysis was carried out to establish what features need to be extracted. Following this, a feature-extracting pipeline was built. During the feature extraction, paths from the dataset that did not have a corresponding image from the cloned meme repository cloned were filtered out. Finally, a logistic regression classifier was built using the LBFS optimiser. To improve the results, the model was trained in multiple folds using the K-Fold algorithm. The different versions of the models generated were then saved. Finally, aggregated scores were generated along with a confusion matrix (select one per cent of the data set alone).

4.2. Convolutional Neural Network

Since CNN (DenseNet121) provided the best results according to the paper, this model was chosen. To build the neural network, a dataset loader was built to provide the interface for loading all the images along with other functionalities. Next, a pre-processing module was built that provided the interface to preprocess all the loaded data and prepare it to feed it to the neural network. For the neural network, as discussed above, DenseNet121 was chosen. The last layer of the model (corresponding to the classifier layer) alone was set to be modified. All the remaining layers were frozen. Finally, the necessary code for training the model was run followed by code to test the model. To evaluate the performance, relevant scores and a confusion matrix were generated.

5. Results

Despite the lack of data and no optimisation of the models, the obtained results almost approach the results as published in the base paper. As described earlier the matrix used to gauge all the models are Mathew’s Correlation Coefficient, the Cohen Kappa score, the F1 Score, and a Confusion Matrix.

5.1. Multinomial Logistic Regression

The three score-base metrics obtained were as follows:

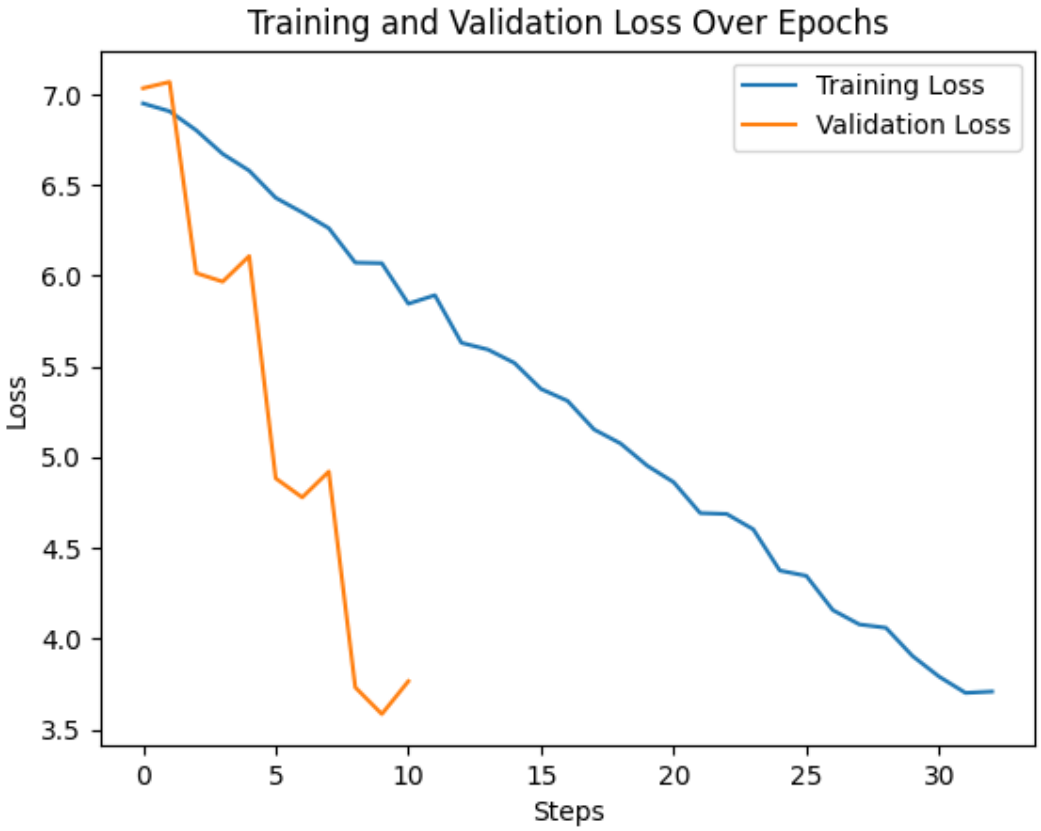
Metric	Mean Score
Mathew's Correlation Coefficient	0.9156
Cohen Kappa Score	0.9156
F1 Score	0.9134

5.2. Convolutional Neural Network

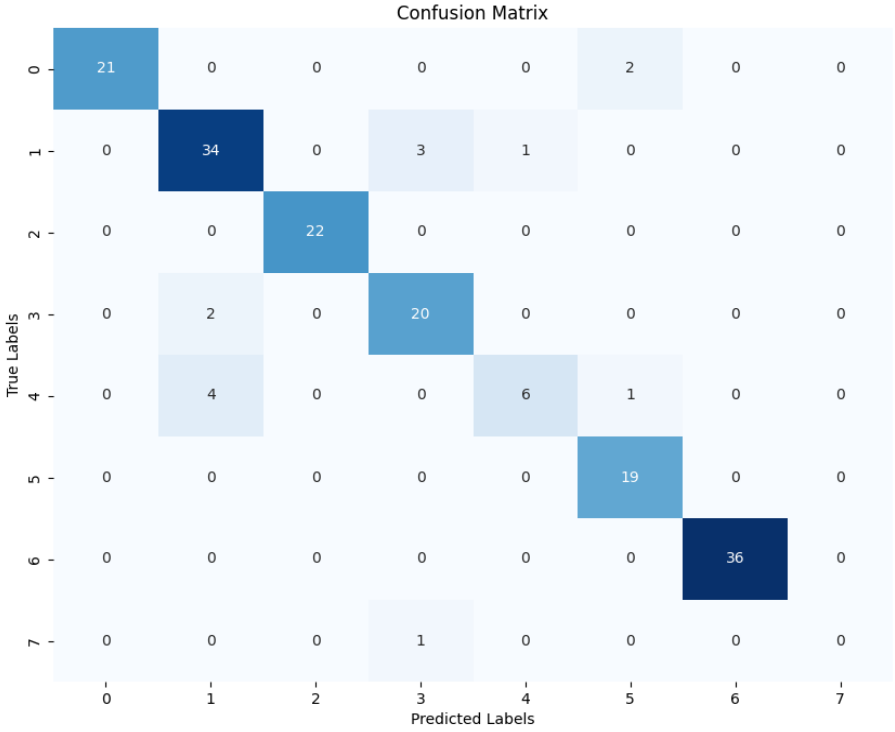
The score-based metrics obtained were as follows:

Metric	Score
Accuracy Score	0.92
Precision Score	0.92
Recall Score	0.92
F1 Score	0.91

The loss graph for the CNN was as follows:



The confusion matrix was as follows:



6. Discussions

6.1. Shortcomings

The primary point of discussion is that even within the two models selected from the paper, the work could not be completely replicated as all the meme data was not available. The work done by the authors used a much more elaborate data set comprising memes from different sources to achieve better results. However, even with the limited amount of image data, the results produced were satisfactory.

This project was originally planned with three models, two from supervised learning and one from unsupervised learning. Owing to time and resource constraints, however, the third model a Hierarchical Density-Based Spatial Clustering of Applications with Noise (H-DBSCAN) model was dropped.

Finally, due to the same reasons as mentioned earlier, the neural network could only be trained with just 5% of the data set, with three epochs. This was mainly done to arrive at some form of results. However, the model trained under CNN is not very efficient, owing to the underfitting.

6.2. Learning Outcomes

This project marked an introduction to practical work in the field of machine learning. It allowed going over different models under both supervised and unsupervised learning. The project also successfully provided a hands-on experience with working on a problem statement relevant to real life.

6.3. Future work

The different models, that were tried out by the authors of the base paper, are very promising when it comes to research in the field of memes. This project aimed to replicate a fraction of the published work within certain constraints. Meme template classification is the stepping stone from which further research can be carried out in this domain.

References

1. L. Murgás, M. Nagy, K. Barnes, and R. Molontay, “Decoding Memes: A Comparative Study of Machine Learning Models for Template Identification,” arXiv.org, 2024.
2. G. Bradski, “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools, 2000.
3. M. Pietikainen, “Local binary patterns,” ¨ Scholarpedia, vol. 5, no. 3, p.9775, 2010.
4. D. S. Evan Klinger, “phash, the open source perceptual hash library,” https://www.phash.org, 2008, accessed: 2024-04-25.
5. A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark et al., “Learning transferable visual models from natural language supervision,” in International Conference on Machine Learning. PMLR, 2021, pp. 8748–8763.
6. L. McInnes, J. Healy, and S. Astels, “hdbscan: Hierarchical density based clustering.” Journal of Open Source Software, vol. 2, no. 11, p. 205, 2017.
7. C. Kwak and A. Clayton-Matthews, “Multinomial logistic regression,” Nursing research, vol. 51, no. 6, pp. 404–410, 2002.
8. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
9. https://github.com/ghost-1608/Meme-Template-Classification