

**Faculty of Informatics and Computer Science**

*Artificial Intelligence*

**PIC: Personal Images Classifier**

By: Ashraf Adel Gamil Haress

Supervised By Associate Professor Nahla Barakat

**November 2022**

**Abstract**

Managing one’s photos is often a monotonous task; one must browse through images in their storage device to distinguish personal images (PIs) from irrelevant ones (IRIs) and keep only the former. To avoid this repetitive process, this project proposes an efficient method utilizing deep and/or other machine learning techniques that will automatically classify the images in question into relevant and irrelevant ones, where the final methodology used for this task is called *PIC: Personal Images Classifier*.

Table of Contents

[1 Introduction 1](#_Toc120283631)

[1.1 Overview 1](#_Toc120283632)

[1.2 Problem Statement 1](#_Toc120283633)

[1.3 Scope and Objectives 1](#_Toc120283634)

[1.4 Report Organization (Structure) 2](#_Toc120283635)

[1.5 Work Plan (Gantt Chart) 2](#_Toc120283636)

[2 Related Work (State-of-The-Art) 3](#_Toc120283637)

[2.1 Background 3](#_Toc120283638)

[2.2 Meme Clustering 3](#_Toc120283639)

[2.2.1 Memes as Features Extracted from Text-Only Posts 3](#_Toc120283640)

[2.2.2 Memes as Features Extracted from Image and Text 4](#_Toc120283641)

[2.2.2.1 Using DeepCluster 4](#_Toc120283642)

[2.2.2.2 Using pHash 5](#_Toc120283643)

[2.2.3 Memes as Features Extracted from Image Only 7](#_Toc120283644)

[2.3 Meme Classification 9](#_Toc120283645)

[2.3.1 Using Visual Features on ML/NN Techniques 9](#_Toc120283646)

[2.3.2 Using Visual and Textual Features on Multi-Modal Models 10](#_Toc120283647)

[2.3.3 Using Hierarchical Image Classification on Multi-Modal Models 12](#_Toc120283648)

[2.4 Face Recognition 14](#_Toc120283649)

[2.5 Analysis of the Related Work 15](#_Toc120283650)

**List of Figures**

Figure 1. Gantt Chart for the PIC project. 2

Figure 2. Projection of proto-memes based on their features that inform how they are similar. In the given example, the tweet content is the most similar aspect of the two proto-memes. Adapted from [1]. 3

Figure 3.t-SNE projection of IRA (i.e., political) and Reddit (i.e., non-political) memes, where each colour indicates a cluster on the embedding space (i.e., feature vectors) learned from DeepCluster [2]. 5

Figure 4.The processing pipeline used by Zannettou et al. [4]. 6

Figure 5.Variants of Smug Frog meme where the computed pHash values for these images are 55352b0b8d8b5b53, 55952b0bb58b5353, and 55952b2b9da58a53 respectively. Adapted from [4]. 6

Figure 6.Visualization of the clusters from fringe web communities [4]. 7

Figure 7.Flow diagram for the creation of a meme influence graph [8]. 8

Figure 8.Left image classified as meme, while middle and right images are not. Adapted from [8]. 8

Figure 9.No-meme, sticker, and meme as classes of images [12]. 9

Figure 10.Left and right plots are the data distribution before and after under sampling [12]. 10

Figure 11.Samples of non-meme and meme images. Adapted from [13]. 10

Figure 12.SN using VGG-16 and Sentence Encoder models, such that Dense 500 is applied on both models’ outputs in order to calculate the Euclidean distance [13]. 11

Figure 13.CCA using feature vectors from VGG-16 and Sentence Encoder models, such that the final highly canonically correlated features are passed to SVM classifier [13]. 12

Figure 14.Semantic class hierarchy [16]. 12

Figure 15. Multi-Modal Architecture [16]. 13

**List of Tables**

[Table 1. Summary of the Related work’s Architectures, where HC: Hierarchical Classification, “-”: the information was not provided by the author, “IF”: Image features, “TF”: Text Features, “FR”: Face Recognition 15](#_Toc120283666)

# Introduction

The following sub-sections elaborate on the classification problem that this project aims to solve.

## Overview

Ever since the inception of mobile storage devices, people have been filling their devices with personal images (PIs) that include family, friends, or selfies. In addition to PIs, these devices have also been cluttered with irrelevant images (IRIs) such as internet memes, and images celebrating a specific event like a holiday or a birthday, which we will refer to as occasional images (OIs). However, IRIs are subjective; one may see screenshots of chat messages as relevant, therefore PIs, while the other thinks they are IRIs. Therefore, we’ll first subjectively define what constitutes as PI and IRI, then discuss the methodology that classifies images based on the established criteria.

## Problem Statement

Given certain criteria that defines PIs and IRIs, automatically classify images into one of the two categories.

## [Scope](http://www.cs.stir.ac.uk/~kjt/research/conformed.html) and Objectives

We will limit the scope of our project by establishing the following criteria for image classification:

1. Personal images (PIs) contain the following:
   1. Family
   2. Friends
   3. Selfies
2. Irrelevant images (IRIs) contain the following:
   1. Internet memes
   2. Social media posts
   3. Chat screenshots
   4. Occasional images. For example, celebrating a certain holiday or birthday.

Therefore, the objective is to establish a methodology using Deep learning and/or machine learning techniques to automate the classification of images based on the specified criteria.

## Report Organization (Structure)

In section 2, we will mention the related work done on subsets of the criteria used to classify images, namely face recognition and meme detection. In section 3, we will elaborate on the proposed solution; the detailed architecture related to pre-processing and classification of images. In section 4, we will state the conclusion of our established methodology and any findings inferred while working on the project.

## Work Plan (Gantt Chart)

Figure 1 represents the Gantt Chart for the project.

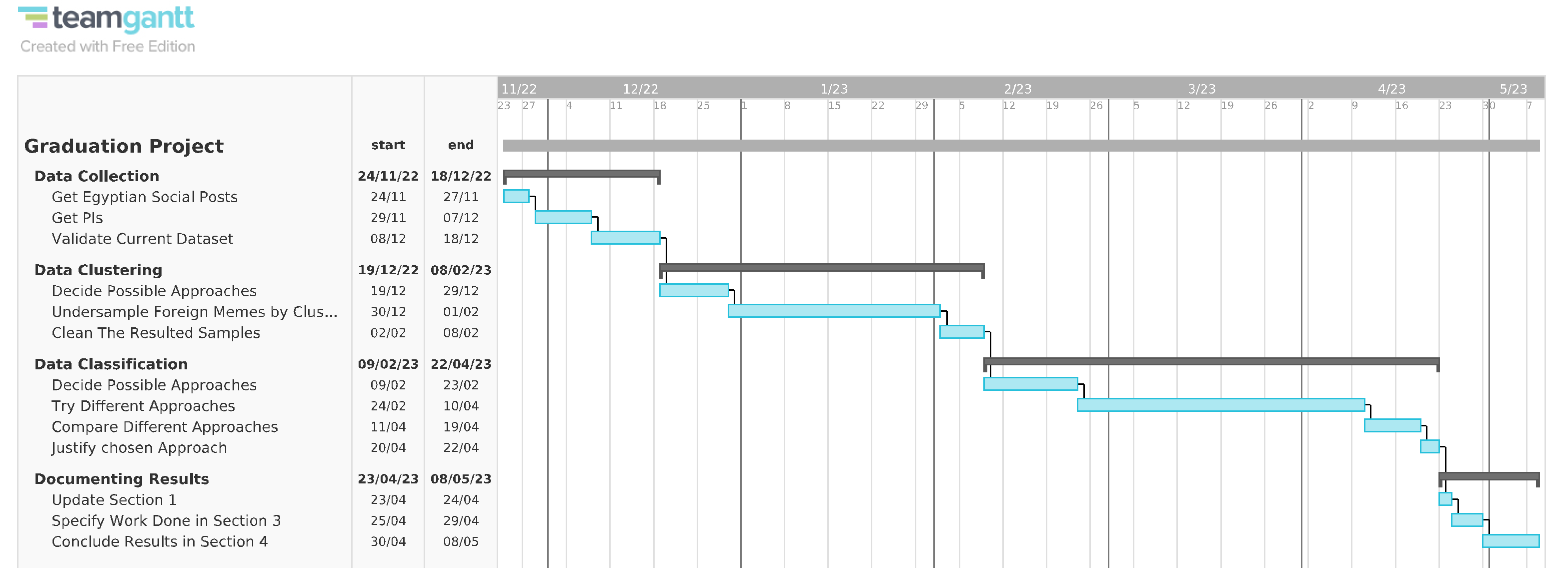


Figure 1. Gantt Chart for the PIC project.

# Related Work (State-of-The-Art)

This section will review previous work done and the state of the art of deep and/or machine learning models related to the meme classification and facial recognition problem.

## Background

Due to the exponential increase of online interactions between internet users, there has been rapid research developments on automatically analysing online content, especially memes sent on various social media platforms. However, since there is no standard format on how memes should look like, researchers re-define what constitutes a meme, based on the problem that they are aiming to solve. For example, Ferrara et al. [1] consider a meme as similar tweets (i.e., message streams) that are clustered together based on similarity, while Chang [2] only considers memes as visual images with superimposed text.

In general, there are common techniques used to solve online-content clustering and classification problems, which are relevant to the problem presented in this project and therefore will be discussed in the upcoming sections.

## Meme Clustering

### Memes as Features Extracted from Text-Only Posts

Ferrara et al.’s research in [1], which is one of the earliest research about memes, focused on clustering grass rooted memes, which are memes that have a role in orchestrated political campaigns, based on textual features such as the hashtag and the post’s text but **not including the post’s image**. The authors refer to these features as *protomemes*. In essence, their proposed clustering framework used similarity measures over proto-memes to aggregate them into broader memes. ‎Figure 2 shows that these similarity measures are defined by considering the distance between the two protomeme’s corresponding features.

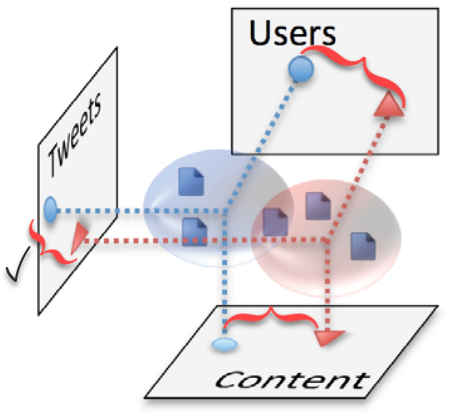


Figure 2. Projection of proto-memes based on their features that inform how they are similar.   
In the given example, the tweet content is the most similar aspect of the two proto-memes.  
Adapted from [1].

Moreover, they compared different strategies of combining different similarity measures, and found that the *pairwise maximization strategy* is the most effective strategy (i.e., obtained the best results statistically), where this strategy chooses the similarity measure with the highest value for every two compared proto-memes and based on this chosen measure, it decides the overall similarity between the two proto-memes.

Regarding the clustering algorithm, they compared *hierarchical clustering* with *K-means clustering* algorithms. Furthermore, to determine the similarity among two clusters, the authors adopted the *average-linkage­* method, which is the average distance between all pairs, such that each pair consists of a proto-meme from the first and second cluster.

Hierarchical clustering is designed to span a range of granularities, where granularity means the level of details of each cluster, so whether it consists of tightly or broadly related proto-memes is based on changing the similarity threshold to produce varying number of clusters. Meanwhile, K-means clustering is more computationally efficient if the desired number of clusters is known in advance.

The result of this comparison shows that K-means performs better only when the number of clusters is relatively few (less than 100), while the hierarchical clustering algorithm has overall better performance over average and large number of clusters. Thus, they use hierarchical clustering for the rest of their experiments.

### Memes as Features Extracted from Image and Text

The following sub-sections discuss two feature extraction methods based on both image and text of a meme.

#### Using DeepCluster

More recently, Chang [2] also analysed and clustered political memes vs non-political ones, but unlike Ferrara et al., he did so by using *DeepCluster* [3]. DeepCluster is a self-supervised architecture that does the following:

1. Take unlabelled images and augment them.
2. Use a Convolutional Network (*ConvNet*) architecture, for example AlexNet or VGG-16, to extract features from the augmented images. Noting that in this case, the author chose VGG-16
3. Use *Principle Component Analysis* *(PCA)* to reduce the dimensions of the feature vector.
4. Pass the reduced feature vector to *K-Means* clustering algorithm to assign each image (in feature vector form) to a cluster.
5. Generate *pseudo-labels* for each feature vector from these cluster assignments.
6. Train the ConvNet architecture to predict these clusters and update its performance by minimizing the multinomial logistic loss (i.e., cross entropy) using mini-batch gradient descent and backpropagation to compute the gradient.

Regarding the implementation of K-means, Chang set the number of clusters hyper-parameter (K) to 100 and used Euclidean distance to cluster the feature vector representation of the images. Based on the architectural setup mentioned above, he was able to visualise the difference between political and non-political (i.e., authentic) memes as shown in ‎Figure 3.

A picture containing text

Description automatically generated

Figure 3.t-SNE projection of IRA (i.e., political) and Reddit (i.e., non-political) memes, where each colour indicates a cluster on the embedding space (i.e., feature vectors) learned from DeepCluster [2].

Regarding step 6. In the DeepCluster architecture, he used logistic regression to obtain a classification F1-score of 0.84.

#### Using pHash

Focusing on the nature of how memes propagate on social platforms, Zannettou et al. [4] grouped various types of memes (including political ones) into clusters to gain insights about what types of memes are present in each social platform, and the influence of such platforms on each other in terms of propagating memes. More specifically, the steps done to group images into clusters and then check which of these clusters have similar images to assign them to higher level groups are illustrated in ‎Figure 4.

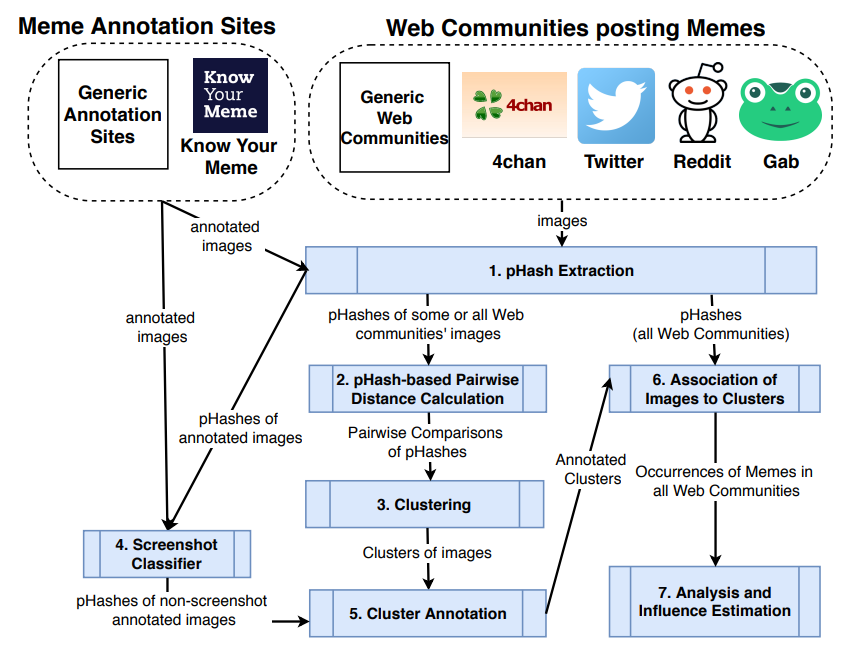


Figure 4.The processing pipeline used by Zannettou et al. [4].

They collected images of memes from various mainstream platforms and fringe web communities, then they used the *Perceptual Hashing (pHash)* algorithm [5] to obtain a fingerprint of each image so that visually similar images map to similar hash values by computing the discrete cosine transform to transform the image from spatial domain into frequency domain, where arithmetic operations are applied on the latter domain to obtain the pHash value. An example of pHashing is provided in ‎Figure 5.



Figure 5.Variants of Smug Frog meme where the computed pHash values for these images are 55352b0b8d8b5b53, 55952b0bb58b5353, and 55952b2b9da58a53 respectively. Adapted from [4].

They then used a custom distance metric for the *DBSCAN* algorithm [6] [7] that is used to cluster the images. This metric uses the following features as similarity measures:

1. Perceptual: It is the similarity of images from a perceptual viewpoint, and it is calculated by getting the Hamming distance, which is the total count of 1 bit in the XOR result of the 2 medoid pHashed images of the clusters that are being compared.
2. Meme, culture, people: They refer to the meme’s given name, associated culture, and people included in the meme respectively.

Noting that only the former feature is used as the distance metric if one or both medoid memes are not annotated from generic annotation sites like know your meme (KYM), while all features are used if both medoid memes are annotated.

After using DBSCAN on their dataset, they obtained clusters visualized in Figure 6.

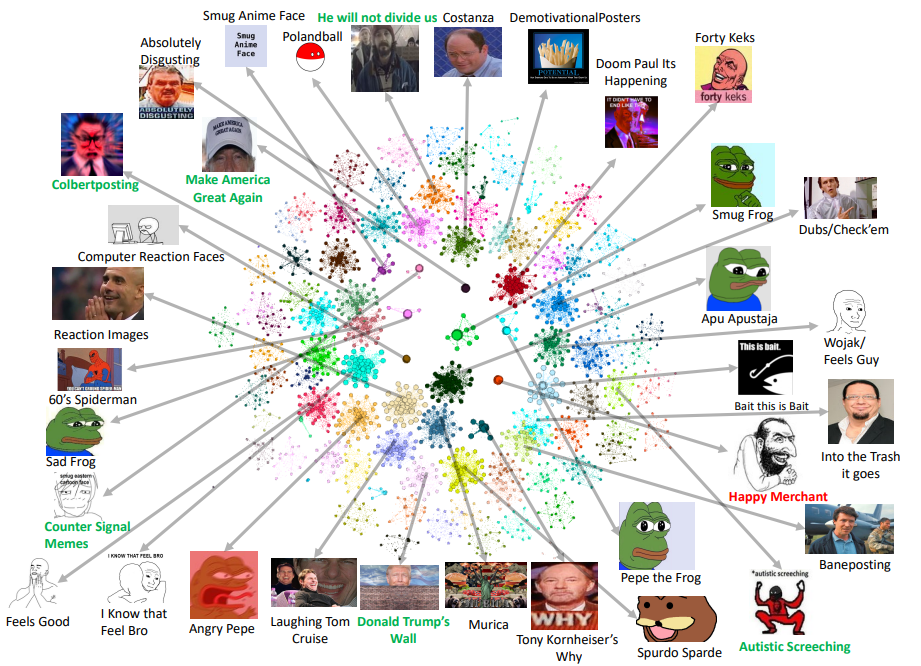


Figure 6.Visualization of the clusters from fringe web communities [4].

After obtaining the clusters, they found insights about the nature of memes and their propagation from these both fringe web communities and mainstream platforms. Finally, they used the Hawkes statistical model to see the social platforms responsible for making certain types of memes viral.

### Memes as Features Extracted from Image Only

Onielfa et al. [8] were also interested in how memes propagate. Unlike Zannettou et al. [4], they focused on specific Instagram users’ influence on other users, instead of entire web communities’ influence. More specifically, Figure 7 illustrates the steps they used for creating a meme influence graph that they used to know the most influential Instagram users, where this information can be leveraged when selecting candidates for marketing campaigns using memes.

Graphical user interface

Description automatically generated

Figure 7.Flow diagram for the creation of a meme influence graph [8].

They first used Instagram’s API to extract posts from 91 users and obtained around 457,101 images. After that, they used a *meme detection algorithm* to discard images with no text or with a large amount of text such that it mostly covers the underlying background image such that 342,984 images (75%) were classified as memes. The algorithm does the following steps:

1. Detect areas containing text using CRAFT text detector [9].
2. Compute text-to-image ratio.
3. If this ratio is not within manually-set lower and upper bounds, then the image has either only text or no text, so the algorithm terminates with the image not being a meme.
4. Otherwise, inpaint the image using Navier–Stokes algorithm [10] such that the text areas obtained from CRAFT are used as inpainting mask.
5. If the standard deviation of the inpainted image’s grayscale values is lower than a manually-set threshold, then there was no content (i.e., meme template) left after removing the text, so the algorithm terminates with the image not being a meme.
6. Otherwise, the algorithm will have extracted the meme template and will terminate with the image being a meme.

Examples of inpainted images classified as memes or not is illustrated in Figure 8.

A picture containing graphical user interface

Description automatically generated

Figure 8.Left image classified as meme, while middle and right images are not. Adapted from [8].

They then used *VGG-16* [11] pre-trained with weights from the ImageNet challenge and its second-to-last fully connected layer to extract a feature vector of 4096 from the text-free meme (i.e., meme template).

After extracting the features, they applied PCA to reduce the dimensionality of these features to 1024, and then they used the resulted features as input to the DBSCAN algorithm which was able to group memes into 13,663 clusters containing 82,801 memes (24%) out of the 342,984 images, where the rest of images were determined by DBSCAN to be noise.

## Meme Classification

Aside from Facebook’s DeepCluster [3] classification step and the meme detection algorithm proposed by Onielfa et al. [8], there are other research that specifically tackled the problem of meme classification.

### Using Visual Features on ML/NN Techniques

Wanting to identify an image as a meme, sticker, or non-meme, Perez et al. [12] tested *Histogram of Oriented Gradient (HOG)* and *ResNet* as feature descriptors to extract features from the image, and applied various classification models on these features, mainly decision tree, K-Nearest Neighbour (KNN), Support Vector Machine (SVM), and Neural Networks (NN). Figure 9 Shows examples of the 3 considered image classes.



Figure 9.No-meme, sticker, and meme as classes of images [12].

Regarding the dataset, they had unequally distributed classes, so they performed under-sampling technique to equally distribute the classes as shown in Figure 10.

Chart, histogram

Description automatically generated

Figure 10.Left and right plots are the data distribution before and after under sampling [12].

Going back to the classification problem, after trying multiple combinations of feature descriptors and learning models, they found that ResNet and Linear SVM had the best results. Therefore, they were able to classify and obtain the images that were considered memes from the dataset from which they were able to implement a system for retrieving memes using textual queries.

### Using Visual and Textual Features on Multi-Modal Models

Relying on both visual and textual elements of an image, Sharma et al., [13] have used a dataset that is created by downloading 20K images from public domains, then they differentiated between meme and non-meme images, where samples are shown in Figure 11.

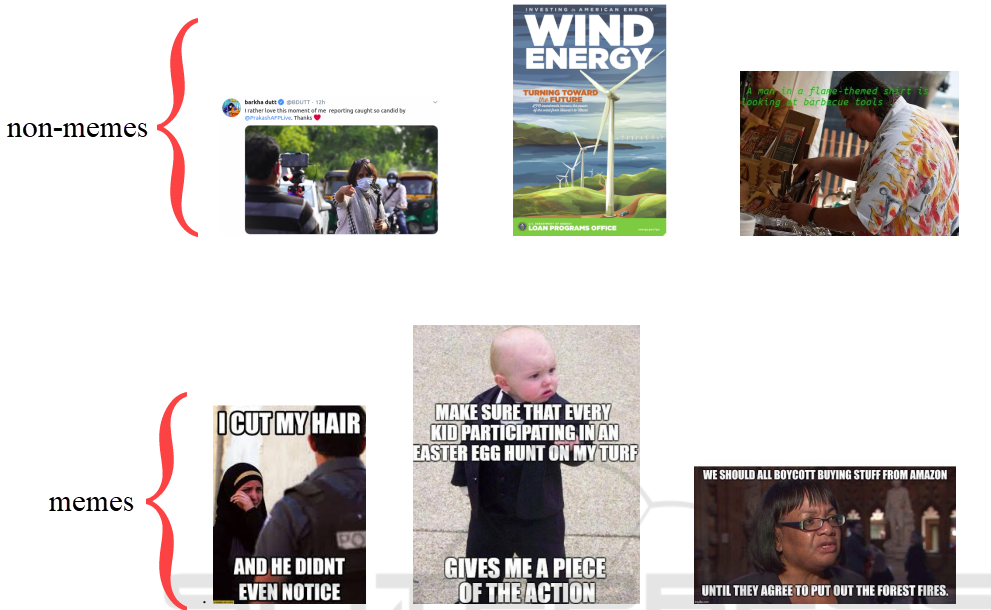


Figure 11.Samples of non-meme and meme images. Adapted from [13].

After preparing the dataset, they did the following steps:

1. For the visual features, they tested both image processing and deep learning techniques on an SVM classifier:
   1. Image processing techniques: HOG, *colour histogram,* *Local Binary Pattern (LBP)*, *Scale Invariant Feature Transform (SIFT)*, *Haar* *technique* which achieved F1-scores of 0.8, 0.56, 0.49, 0.75, and 0.91 respectively.
   2. Deep learning techniques: pre-trained models for ResNet, *AlexNet*, *InceptionNet*, and VGG-16, where the last model achieved the highest F1-score of 0.94. Therefore, VGG-16 was subsequently used to get the visual-based features of an image.
2. For the textual features, they tested *N-Gram, Glove Embedding,* and *Sentence Encoder* and obtained F1-scores of 0.51, 0.9, and 0.95 respectively. They therefore used Sentence Encoder to get the textual-based features of an image.

As can be seen, they have already achieved good scores using an SVM classifier with one of the two modalities (visual or textual). However, due to the multi-modal nature of meme content, they believed that combining these feature descriptors using a multi-modal would yield even better results. They therefore tested the dataset on 2 multi-modal models: *Siamese Network (SN)* [14] and *Canonical Correlation Analysis (CCA)* [15].

Regarding Siamese Network (SN), it checks if two input vectors of the same modality are similar by passing each input vector to a deep learning model, such that the models are equivalent in terms of architecture and weights, and then measuring the similarity according to the Euclidean distance of the resulted 2 feature vectors. However, Sharma et al., [13] applied the theory of SN on two different models: the VGG-16 and Sentence Encoder models, as can be shown in Figure 12.

Diagram

Description automatically generated

Figure 12.SN using VGG-16 and Sentence Encoder models, such that Dense 500 is applied on both models’ outputs in order to calculate the Euclidean distance [13].

Regarding the Canonical Correlation Analysis (CCA) model and in the context of the given visual and textual feature vectors *V* and *T*, the CCA model will find linear combinations of *V* and *T* that maximizes their Pearson correlation value. Then, the highly correlated canonical features are projected into a correlated semantic space and are considered the final feature vector that is passed to an SVM classifier to predict whether an image is a meme or not. The process of CCA is illustrated in Figure 13.

Shape, polygon

Description automatically generated

Figure 13.CCA using feature vectors from VGG-16 and Sentence Encoder models, such that the final highly canonically correlated features are passed to SVM classifier [13].

As for the results, SN and CCA obtained 0.98 and 0.99 F1-scores respectively, which proves, at least for the authors’ case, that classifying memes using multi-modal features yields better results than classifying based on single-modal features.

### Using Hierarchical Image Classification on Multi-Modal Models

While all the aforementioned research is about detecting memes vs non-memes, Das and Mandal [16] focused on memes only, but they also classified the connotation of different types of memes as shown in Figure 14.

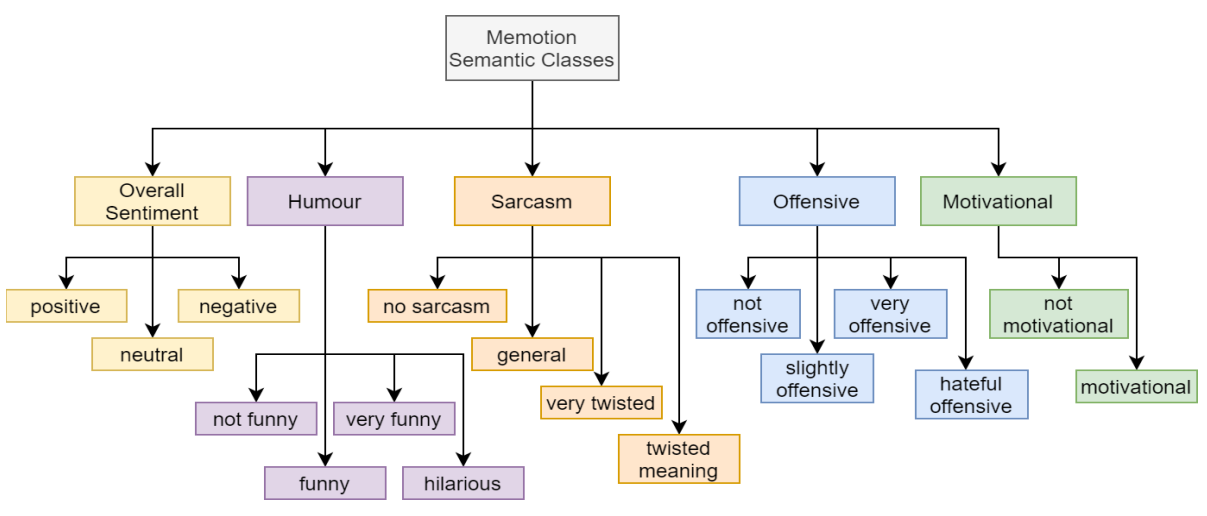


Figure 14.Semantic class hierarchy [16].

Specifically, they accomplished 3 tasks:

1. Task A: To classify if a meme is positive, neutral, or negative (i.e., “Overall Sentiment” sub-labels).
2. Task B: classify humour type (i.e., the other 4 higher labels presented in Figure 14)
3. Task C: classify the scale of each type of the labels from task B (i.e., the sub-labels in Figure 14)

Therefore, they applied a hierarchical classification architecture, such that its structure is expressed in Figure 15.

Diagram

Description automatically generated

Figure 15. Multi-Modal Architecture [16].

For image feature extraction, they first resized the input image, which is from the memotion dataset on Kaggle, into 224x224 sized images. These images are used as input to ResNet for feature extraction. Meanwhile, a recurrent deep neural network (DNN) model consisting of bidirectional *Long-Short-Term-Memory (LSTM)* and *Gated Recurrent Unit (GRU)* is used for text feature extraction. For the text to be used as input in the DNN model, it went through the following steps:

1. Pre-process the text (e.g., converting text to lowercase).
2. Get an initial sub-word token and its embedding by feeding the text into a sub-word tokenizer.
3. Get the vector of sub-word embedding for each token using *SentencePiece Processor* [17].
4. Generate a more detailed embedding vector for each sub-word token by feeding the vectors obtained in the previous step into another trainable embedding layer. That generated vector is now used as input in the DNN model.

The text feature vector resulted from the DNN model is then passed to global average and max pooling layer to get a reduced text feature vector that is finally passed by a dense layer to obtain a further dimensionally reduced text feature vector.

After getting both visual and textual feature vectors, they were concatenated to get an aggregated feature vector that will be used for the multi-label hierarchical meme classification, by predicting the overall sentiment of the meme. That prediction is then concatenated with the aggregated feature vector, using a dense layer with ReLU activation function, to possibly obtain other labels and their fine-grained counterparts using dense layers with sigmoid activation functions for each of the other labels. Refer to Figure 14 and Figure 15 to review the labels and their respective dense layers.

Finally, the model got the following highest scores on tasks A, B, and C respectively:

1. Macro F1-scores: 0.3488 (SE-ResNet18), 0.5112 (ResNet34), 0.3240 (ResNet34)
2. Micro F1-scores: 0.5022, 0.6685, 0.4402 (all with ResNet18)

Where the authors assumed data imbalance and intraclass correlation among fine grained class labels was the cause of poor performance results, especially for task C.

## Face Recognition

Du et al. [18] have done an intensive survey over the methods of each step towards face recognition (FR). However, in this project, we’ll mention only the state-of-the-art models of the face representation step, which is the last step that recognizes faces based on the extracted discriminative features from the image pre-processed from previous steps of the FR steps-pipeline.

Out of the models used for the FR task and based on overall performance on the datasets presented in [18], the *GroupFace* [19] model with backbone of ResNet-100 obtained the best scores between the classification models, while *VGG Face* [20] and *GridFace* [21] achieved the best scores between the embedding models, and *AFRN* [22] had the best scores between the hybrid models. Noting that each subcategory of models means the following:

1. Classification: Considers the face representation learning as a classification task.
2. Feature Embedding: Optimizes the feature distance according to the label of sample pair.
3. Hybrid: Applying the above two subcategories together as the supervisory signals.

## Analysis of the Related Work

Table 1 represents a summary of the discussed methodologies in section 2 and their performance.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Model Type** | **Feature Extraction Method** | **Dataset** | **Performance/Results** |
| Meme Clustering | HC [1] | Proto-memes | Twitter: 5.5K tweets (excluding image) | LFK-NMI better than K-Means when #Clusters 100 |
| K-Means | LFK-NMI better than  HC when #Clusters 100 |
| DBSCAN [6]  (Used by [4]) | pHash | #pHashes:  Twitter: 74M  Reddit: 30M  4Chan: 3.6M  Gab: 193K | Noise / non-noise Ratio[[1]](#footnote-1):  Twitter: -  Reddit: 0.64 (19.2M/30M)  4Chan: 0.63 (2.27M/3.6M)  Gab: 0.69 (133.17K/193K) |
| DBSCAN [6]  (Used by [8]) | VGG-16 | Instagram: 343K Memes | Noise / non-Noise Ratio: 0.76 (260.7K/343K) |
| Meme Classification | Meme Detection Algorithm [8] | None | Instagram: 457K Images | Evaluation Results: -  Classification Results:  342.75K memes,  114.25K non-memes |
| Linear-SVM  (Used by [12]) | ResNet-152 | Twitter Images:  1.2K Memes  1.4K Stickers  49.3K non-Memes | Average F1-score: 0.73 |
| SN [14]  (Used by [13]) | IF: VGG-16  TF: Sentence Encoder | Flicker8K & Twitter Images:  7K Memes  7K non-Memes | F1-score: 0.98 |
| CCA [15]  (Used by [13]) | F1-score: 0.99 |
| Multi-Label Hierarchical Meme Classification | Multi-Modal Architecture of CNN on Aggregated Features [16] | IF: ResNet (SE18, 18, 34)  TF: SentencePiece Processor[17] then DNN of LSTM & GRU | Memotion Dataset: 7K Memes [23] | Average-Macro F1-score: 0.3946  Average-Micro F1-score: 0.53697 |
| Meme Clustering & Classification | DeepCluster [3]  (Used by [2] with LR) | VGG-16 | IRA Memes: 26K  Reddit Memes: 26K | Clustering Evaluation: -  Classification F1-score: 0.84 |
| FR: Classification | GroupFace [19] | ResNet-100 | 10 Face Datasets [18] | Avg Acc: 97.265% |
| FR: Embedding | VGG Face | CNN-36 | Datasets 1-3, 6 | Avg. Acc.: 84.84% |
| GridFace | GoogLeNet-22 | Datasets 1 and 6 | Avg. Acc.: 97.65% |
| FR: Hybrid | AFRN | ResNet-101 | Datasets 1, 4-10 | Avg. Acc.: 95% |

Table 1. Summary of the Related work’s Architectures, where HC: Hierarchical Classification, “-”: the information was not provided by the author, “IF”: Image features, “TF”: Text Features, “FR”: Face Recognition

It can be shown that the evaluation metrics for the clustering models are not consistent, as the authors were focusing on inferences from the generated clusters. Meanwhile, above-average scores for classification models were achieved, except for the multi-modal architecture. However, it is important to state that research on hierarchical meme classification is not as extensive as research on single-level meme classification.

**References**

[1] E. Ferrara, M. JafariAsbagh, O. Varol, V. Qazvinian, F. Menczer, and A. Flammini, “Clustering memes in social media,” in *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*, Aug. 2013, pp. 548–555. doi: 10.1145/2492517.2492530.

[2] K.-C. Chang, “Mapping Visual Themes among Authentic and Coordinated Memes.” arXiv, Jun. 05, 2022. doi: 10.48550/arXiv.2206.02306.

[3] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep Clustering for Unsupervised Learning of Visual Features.” arXiv, Mar. 18, 2019. doi: 10.48550/arXiv.1807.05520.

[4] S. Zannettou *et al.*, “On the Origins of Memes by Means of Fringe Web Communities.” arXiv, Sep. 22, 2018. Accessed: Nov. 22, 2022. [Online]. Available: http://arxiv.org/abs/1805.12512

[5] V. Monga and B. L. Evans, “Perceptual Image Hashing Via Feature Points: Performance Evaluation and Tradeoffs,” *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3452–3465, Nov. 2006, doi: 10.1109/TIP.2006.881948.

[6] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, Oregon, Aug. 1996, pp. 226–231.

[7] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, “DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN,” *ACM Trans. Database Syst.*, vol. 42, no. 3, pp. 1–21, Aug. 2017, doi: 10.1145/3068335.

[8] C. Onielfa, C. Casacuberta, and S. Escalera, “Influence in Social Networks Through Visual Analysis of Image Memes,” 2022. doi: 10.3233/FAIA220317.

[9] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, “Character Region Awareness for Text Detection.” arXiv, Apr. 03, 2019. doi: 10.48550/arXiv.1904.01941.

[10] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, “Navier-stokes, fluid dynamics, and image and video inpainting,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Kauai, HI, USA, 2001, vol. 1, p. I-355-I–362. doi: 10.1109/CVPR.2001.990497.

[11] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition.” arXiv, Apr. 10, 2015. doi: 10.48550/arXiv.1409.1556.

[12] J. Perez-Martin, B. Bustos, and M. Saldana, “Semantic Search of Memes on Twitter.” arXiv, May 20, 2020. doi: 10.48550/arXiv.2002.01462.

[13] C. Sharma, V. Pulabaigari, and A. Das, “Meme vs. Non-meme Classification using Visuo-linguistic Association,” in *Proceedings of the 16th International Conference on Web Information Systems and Technologies*, Budapest, Hungary, 2020, pp. 353–360. doi: 10.5220/0010176303530360.

[14] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese Neural Networks for One-shot Image Recognition,” p. 8, 2015.

[15] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, “Canonical Correlation Analysis: An Overview with Application to Learning Methods,” *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, Dec. 2004, doi: 10.1162/0899766042321814.

[16] S. D. Das and S. Mandal, “Team Neuro at SemEval-2020 Task 8: Multi-Modal Fine Grain Emotion Classification of Memes using Multitask Learning.” arXiv, May 21, 2020. Accessed: Nov. 23, 2022. [Online]. Available: http://arxiv.org/abs/2005.10915

[17] T. Kudo and J. Richardson, “SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing.” arXiv, Aug. 19, 2018. doi: 10.48550/arXiv.1808.06226.

[18] H. Du, H. Shi, D. Zeng, X.-P. Zhang, and T. Mei, “The Elements of End-to-end Deep Face Recognition: A Survey of Recent Advances.” arXiv, Dec. 27, 2021. Accessed: Nov. 23, 2022. [Online]. Available: http://arxiv.org/abs/2009.13290

[19] Y. Kim, W. Park, M.-C. Roh, and J. Shin, “GroupFace: Learning Latent Groups and Constructing Group-based Representations for Face Recognition.” arXiv, May 25, 2020. doi: 10.48550/arXiv.2005.10497.

[20] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep Face Recognition,” in *Procedings of the British Machine Vision Conference 2015*, Swansea, 2015, p. 41.1-41.12. doi: 10.5244/C.29.41.

[21] E. Zhou, Z. Cao, and J. Sun, “GridFace: Face Rectification via Learning Local Homography Transformations.” arXiv, Aug. 19, 2018. doi: 10.48550/arXiv.1808.06210.

[22] B.-N. Kang, Y. Kim, B. Jun, and D. Kim, “Attentional Feature-Pair Relation Networks for Accurate Face Recognition.” arXiv, Aug. 17, 2019. doi: 10.48550/arXiv.1908.06255.

[23] W. Scott, “Memotion Dataset 7k,” 2020. https://www.kaggle.com/datasets/abe12cc1fbe90bb29b63a0438028e1a7c989789bf217c9d54dc05aca186cbd4c (accessed Nov. 24, 2022).

1. Noise refers to the percentage of images not clustered. Zannettou et al. [4] state that these images are “one-off” images that are not considered memes. [↑](#footnote-ref-1)