UNIVERSITAT DE BARCELONA

**DEGREE IN MATHEMATICS**
**DEGREE IN COMPUTER SCIENCE**

**Bachelor's Thesis**

---

# Visualizing and Ranking the Influence of Users who Post Memes in Social Networks

---

**Author: Carles Pérez Onielfa**

**Supervisors:** Dr. Sergio Escalera Guerrero
Dr. Carles Casacuberta Vergés

**Department of Mathematics and Computer Science**
**Barcelona, June 13, 2022**

# Contents

## Abstract

Memes evolve and mutate through their diffusion in social media. They have the potential to propagate ideas and, by extension, products. Many studies have focused on memes, but none so far, to our knowledge, on the users who post them, their relationships, and the reach of their influence. In this project, we define a meme influence graph together with suitable metrics to visualize and quantify influence between users who post memes. Then, we describe a process to implement our definitions using a new approach to meme detection based on text-to-image area ratio and contrast. After applying our method to a set of users of the Instagram platform, we conclude that our metrics add information to already existing user characteristics and that our methodology can also be used to study the popularity of memes types among the users.



## Resumen

Los memes evolucionan y mutan a través de su difusión en las redes sociales. Tienen el potencial de propagar ideas, y por extensión, productos. Se han llevado a cabo estudios enfocados en memes, pero ninguno de ellos, a nuestro saber, sobre los usuarios que los publican, cómo se relacionan y el alcance de su influencia. En este proyecto definimos un grafo de influencia de memes junto con métricas que nos permiten visualizar y cuantificar la influencia entre usuarios que publican memes. A continuación, describimos un proceso para implementar las definiciones proporcionadas a partir de un nuevo enfoque para la detección de memes que se basa en el contraste y en la relación entre las áreas de texto e imagen. Finalmente, aplicamos nuestra metodología a un conjunto de usuarios de la plataforma Instagram y concluimos que nuestras métricas aportan información adicional sobre las características ya existentes de los usuarios y que nuestra metodología también es aplicable para visualizar la popularidad de los tipos de memes entre los usuarios.



2020 Mathematics Subject Classification: 05C22, 62H30, 68T07, 68T45, 91C20

# Chapter 1

# Introduction

A meme is usually defined as “an idea, behavior, phrase or usage that spreads within a culture” [7]. In the digital era, memes have adapted to new technologies and have become a phenomenon in contemporary web culture [21]. As a combination of humor, text, and a symbol, emoticons became one of the first types of Internet memes.

Even though Internet memes can exist as text, emojis, videos, or gifs, a common format is that of an image with superimposed text that conveys some type of message in an epigrammatic style. In the earlier days of the Internet, images with superimposed text began to propagate via e-mail and message boards. Later, social networks emerged, allowing memes to viralize [6]. Image memes have become an integral part of Internet culture. With the help of users they are born and reproduced, often mutated in the process. They are also used to spread political messages and ideologies. Compared to textual memes, image memes can condense their content and require less attention to be understood. Therefore, they are likely to be more effective [24].

Many studies have been carried out around memes, mainly focusing on their evolution [3], predicting their virality [2, 24], modeling their spread with mathematical models [4, 39], or devising algorithms for detecting them [6, 13]. But few, if any, have dug deeper behind the creators of memes.

Regarding human achievement, viral success is closely related to merit [43]. Therefore, it is natural that memes that were once uploaded anonymously are now being uploaded by users that are proud of their creations and sign their memes with their watermark. Some users who post popular memes have achieved massive followings, and this grants them enormous influence and reach. However, that would be true of any user on a social network with a big number of followers. What makes meme creators unique is that they not only have the power to reach their followers, but two factors greatly expand their scope. First, memes are meant to be spread and shared; hence, followers of meme creators, if they enjoy a meme, are likely to share it with their friends [40]. Second —and most importantly— from an original meme, other creators can mutate and alter the original to make their own, retaining core aspects of the meme such as the underlying image. If there was an idea or product within it, as the meme and its mutations viralize and are shared, the idea or product goes viral with it, achieving exposure orders of magnitude greater than the original reach of the creator of the meme. In Figures 1.1 and 1.2 we can find examples of memes featuring images that originate around a product.

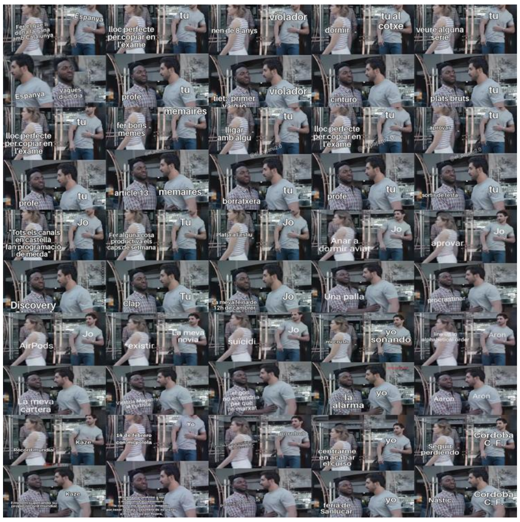


Figure 1.1: Memes containing a frame from Gillette's commercial *We Believe: The Best Men Can Be* (2019).

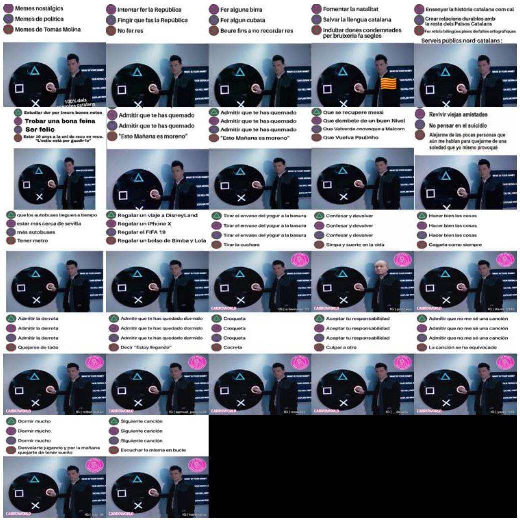


Figure 1.2: Memes containing a frame from the videogame *Detroit: Become Human* (2018).

## 1.1. Objectives

In this study, we take over the task of providing tools to gain insight into creators and the relationship between them through a visual analysis of the content of their memes. More precisely, this project aims to provide the following contributions:

1. A definition of a graph for visualizing relationships between users who post memes on social networks.

2. Metrics based on graph theory that evaluate and rank users concerning meme virality.

3. A process for experimentally building the graph as mentioned earlier with a new approach for detecting image memes.

4. A demonstration of how this process can be applied to a set of Spanish users who post memes on Instagram.

Indirectly, by accomplishing the previous contributions, the following objectives will also be achieved:

1. Building a pipeline for extracting and storing large amounts of data from a social network.

2. Implementing algorithms for detecting memes based on text detection and image contrast.

3. Extracting features from image data using a Convolutional Neural Network.

4. Clustering images based on their underlying images using their extracted features.

5. Interpreting results from applying our methodology to a set of users.

## 1.2. Hypothesis

The following hypothesis can be raised from this project: By building the graph and computing the metrics that we define, we will be able to visualize relationships between users and rank the users regarding the potential of their memes to become viral and influence other users.

We believe that some of these users would be overlooked by using standard metrics for determining influence. As a bonus, by collecting the necessary data to perform this task we will also be able to have an insight into the memes themselves, including their popularity and longevity.

# Chapter 2

# Preliminary Concepts

In this chapter we explain concepts needed to fully understand this study in varying levels of depth depending on their complexity and relevance to the project.

## 2.1. Social Media Platforms

A *social media platform* is any medium where a user is capable of broadcasting content to the general public or a subset of the general public. This content includes images, videos, messages, and sound files. Examples of social media platforms include *YouTube*, *Facebook*, and *Twitter*. Recent statistics estimate that as of April 2022 there are 4.65 billion social media users [20] totaling 58.7 percent of the global population.

### 2.1.1. Instagram

Instagram is a social media platform focused exclusively on photo and video sharing. Users can take, edit and publish visual content on their profile, which can be browsed by any other user (if the user's profile is public), or only their followers (if the profile is private). As a user, you can follow any other public user. By following a user, you make it so whenever you browse Instagram's main page, you will be shown the recent content uploaded by that user.

On this platform, a post can contain an image, a video, or a heterogeneous collection of images and videos (*albums*). They can also have a short text description which can include tags (*hashtags*) and a location assigned by the user. Users can publicly engage with the posts via likes and comments, and privately by sharing the post with another user or bookmarking (saving) the post. Other features that Instagram includes are messaging, live streaming, reels, and stories, but these are not relevant to this study.

In the implementation of this study, we limited the scope to Instagram. Since Instagram only allows users to publish images and videos, it is likely to find users whose content is mainly image memes. Furthermore, Instagram is the third biggest social media platform [20] and, on this platform, it is common for brands to partner with influential users (influencers) and publish sponsored posts [10]. Thus, metrics for determining influential users are valuable.

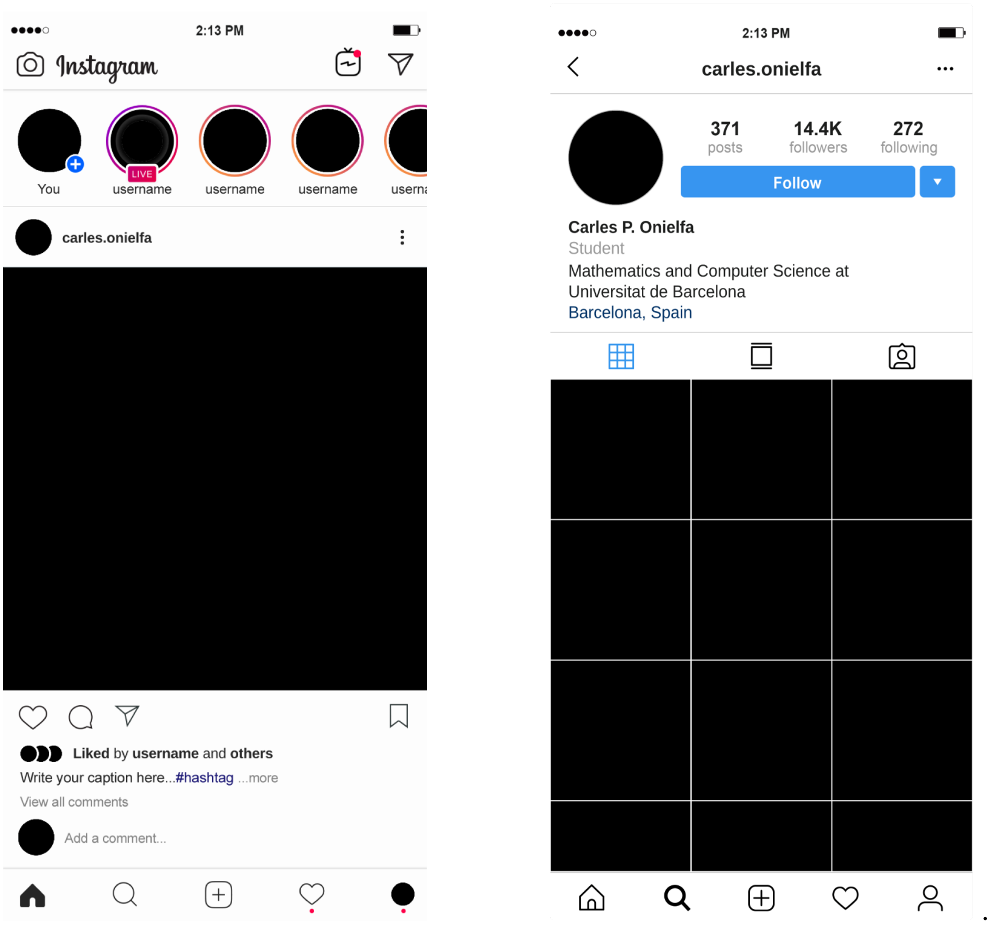
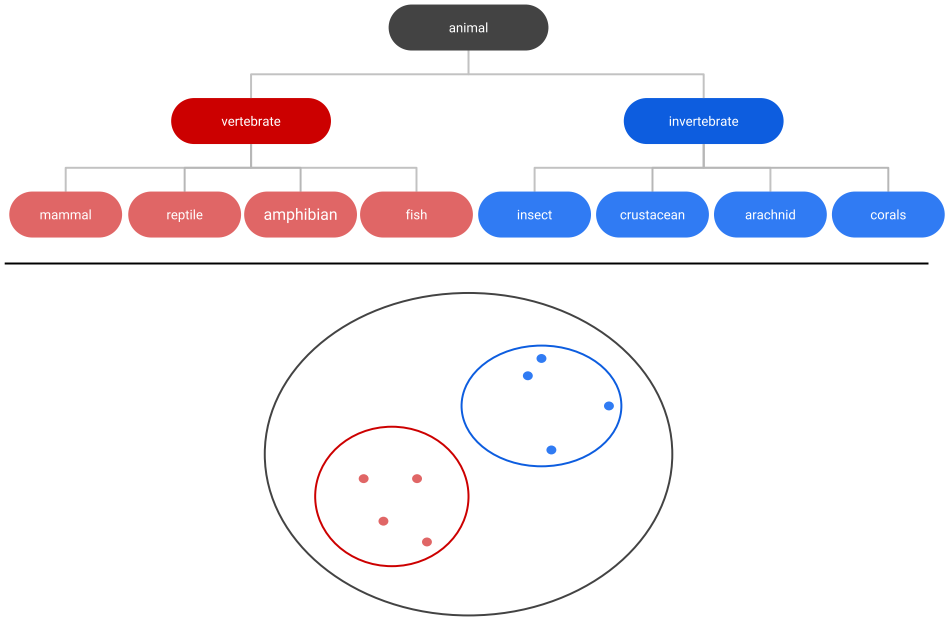


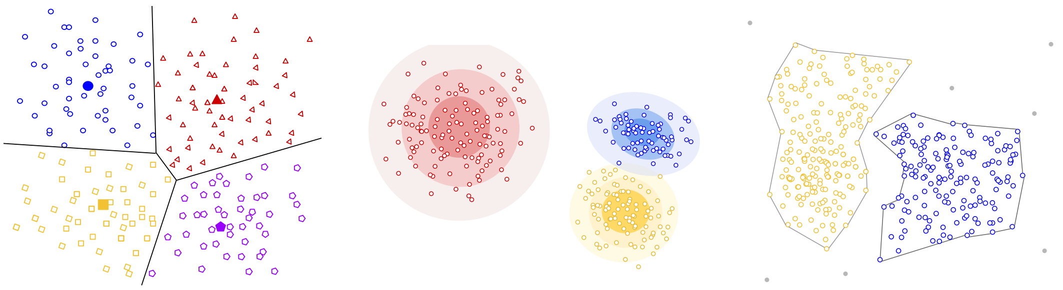

Figure 2.1: Instagram's main page (left) and profile page (right)

## 2.2. Clustering

Clustering is the procedure of grouping a set of vectors (data points) based on a distance function. A clustering algorithm has a set of vectors and a distance function as inputs, and it outputs an integer for each input vector. The integer assigned to each vector denotes what group the vector belongs in. Clustering al-




(a) Hierarchical Clustering.



(b) Centroid-based Clustering.

(c) Distribution-based Clustering.

(d) Density-based Clustering.

Figure 2.2: Examples of the described types of clustering. Extracted from [16].

gorithms can be divided into several categories. As an example here we describe four common types of clustering algorithms (Figure 2.2). More detailed information and an expanded list of categories can be found in [41].

1. **Centroid-based Clustering** groups the data points by the closeness of a data point to the centroid of the cluster. A popular algorithm that falls in this category is $K$-means clustering.

2. **Hierarchical Clustering** assumes that the data follows a hierarchical structure. Two approaches to this type of algorithm are Agglomerative Cluster-

ing, where each data point starts as its own cluster, and then the clusters are merged based on the distance between the clusters; and Divisive Clustering, where we start with one cluster containing all the data points and then we iterate by dividing the cluster.

3. **Distribution-based Clustering** assumes that the data follows a known distribution such as a Gaussian distribution. The probability of a data point belonging to a cluster increases the closer it is to the center of the distribution.

4. **Density-based Clustering** creates clusters from high density areas. Data points that are in areas of low density are classified as outliers and not assigned to any cluster.

### 2.2.1. DBSCAN

**Density-Based Spatial Clustering of Applications with Noise** (DBSCAN) is the algorithm chosen for performing the clustering in this project. DBSCAN is a density-based clustering algorithm proposed in [14]. For DBSCAN clustering, the points are divided into *Core Points*, *Reachable Points*, and *Outliers/Noise*.

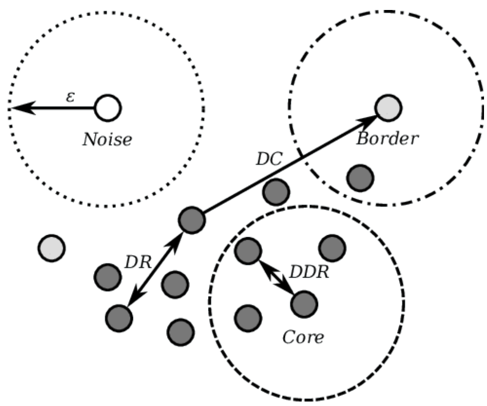



Figure 2.3: Example of DBSCAN clustering with $minPts = 4$, extracted from [17].

The criteria for assigning a category to each point are:

1. A point $p$ is a *core point* if there are more than $minPts$ points within an $\epsilon$ radius.

2. A point $q$ is *directly density reachable (DDR)* from a core point $p$ if the distance between $q$ and $p$ is lower than $\epsilon$.

3. A point $q$ is *density reachable (DR)* from another point $p$ if there is a path $p = p_0, p_1, \ldots, p_n = q$ where each $p_{i+1}$ is directly density reachable from $p_i$.

4. Two points $p$ and $q$ are *density-connected (DC)* if there exists a point $o$ such that $p$ and $q$ are reachable from $o$.

5. All other points are *outliers*.

Clusters are then defined as groups of points where all the points are mutually density-connected and contain every density-reachable point from another point in the cluster (Figure 2.3).

## 2.3. Graphs

Concepts related with graphs will be detailed in this section, since, although basic, they are essential for this study.

A *directed graph* $G = (V, E)$ consists of a non-empty finite set $V$ whose elements are called *vertices* (or *nodes*) and a finite set $E$ of distinct ordered pairs of distinct elements of $V$ called *edges*. An edge $(v, w)$ is said to *join* the vertices $v$ and $w$. A representation of an example of a directed graph is shown in Figure 2.4.

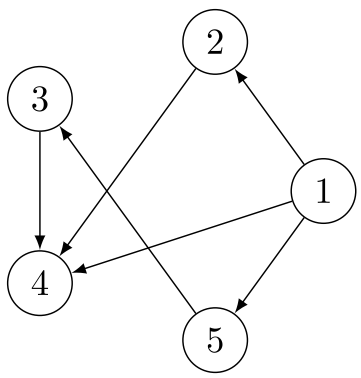



Figure 2.4: Representation of a directed graph $(V, E)$ with $V = \{1, 2, 3, 4, 5\}$ and $E = \{(1, 2), (1, 4), (1, 5), (2, 4), (3, 4), (5, 3)\}$.

Let $(V, E)$ be a directed graph. If a function $w \colon E \longrightarrow \mathbb{R}^+$ is given, we call it a *weight function* and say that $(V, E, w)$ is a *directed weighted graph*.

From now on, let $G = (V, E, w)$ be a directed weighted graph with vertices $V = \{v_1, \dots, v_n\}$. The *adjacency matrix*, $A$, of $G$ is a square $n \times n$ matrix:

$$A = (a_{i,j}),\ i, j \in \{1, \dots, n\}, \text{ with } a_{i,j} = w(v_j, v_i).$$

The following metrics will be relevant for the formalization of the main problem in this project:

- The *out-degree* of a vertex $v_i$ is the number of outgoing edges from $v_i$:

$$d_{\text{out}}(u_i) = \#\{e \in E \mid e = (u_i, u_j),\ i, j \in \{0, \ldots, n\},\ i \neq j\}.$$

- The *in-degree* of a vertex $v_i$ is the number of incoming edges to $v_i$:

$$d_{\text{in}}(u_i) = \#\{e \in E \mid e = (u_j, u_i),\ i, j \in \{0, \ldots, n\},\ i \neq j\}.$$

- The *weighted out-degree* of a vertex $v_i$ is the sum of the weights of the outgoing edges from $v_i$:

$$wd_{\text{out}} = \sum_{\substack{j=0 \\ j \neq i}}^{n} w(u_i, u_j).$$

- The *weighted in-degree* of a vertex $v_i$ is the sum of the weights of the incoming edges to $v_i$:

$$wd_{\text{in}} = \sum_{\substack{j=0 \\ j \neq i}}^{n} w(u_j, u_i).$$

A *community* of a graph is a subset of vertices that are densely connected to each other. For example, if our graph represents social connections, the communities in this graph will be a representation of social circles.

### 2.3.1. PageRank

The *PageRank* algorithm applied to a directed graph measures the importance of each of its vertices taking into account the number of incoming edges and the importance of the source vertices of these edges. In short, a vertex will be important if other important vertices link to it. It was first used to rank web pages in the Google search engine and it was first defined in [36]. Most often this algorithm is described with a random walk approach, in which the PageRank value for a vertex $v_i$ is interpreted as the probability that by infinitely and randomly navigating the edges of the graph we end up in $v_i$.

To describe the probability of transitioning from one vertex to another, the *transition matrix* $M$ is used. It is defined as $M = AD^{-1}$ where $D = \text{diag}(\{\sum_j a_{i,j}\})$. The transition matrix can be thought of as a column-normalized adjacency matrix. The sum of each of the columns in $M$ equals one and every element is non-negative. Thus, $M$ is *column stochastic*.

Let $p^k = (p_1^k, \ldots, p_n^k)$ be the probability of being at each of the vertices after taking $k$ steps, with $p^0 = \left(\frac{1}{n}, \ldots, \frac{1}{n}\right)$. By iteratively multiplying with the transition matrix we obtain a *Markov chain* with which we can obtain $p^k$ at every step. After infinitely many steps, we have:

$$p^{k+1} = Mp^k \xrightarrow{k\to\infty} p = Mp.$$

From this, we can see that $p$ is an eigenvector for the eigenvalue 1. In practice, this eigenvector can be computed using the *power method*, which is equivalent to computing the Markov chain until the value of $p$ converges.

This formulation of the PageRank algorithm has issues. First, nodes that do not have any outgoing edges will set the probability of reaching any other page to 0. Second, the graph can exhibit cycles that make the *power method* diverge, and third, when the graph is formed by more than one connected component there will be multiple linearly independent eigenvectors of eigenvalue 1. The first issue is solved by adding virtual edges from the problematic node to all the nodes in the graph. To address the second and third issues, the following reformulation of the transition matrix was proposed in [36]:

$$\widetilde{M} = (1-d)M + \tfrac{d}{n}\, I.$$

Here $\widetilde{M}$ is referred to as the *Google matrix* and $d$ is the *damping factor*, set to 0.15. The Google matrix fixes the aforementioned issues by allowing for transitions from any vertex to any other vertex with probability $d$.

If $A$ is the adjacency matrix for a graph $(V, E, w)$, the *reverse PageRank* of the node $v$ is the value that the PageRank algorithm for the graph with adjacency matrix $A^t$ (transpose of $A$) assigns to $v$. By computing the PageRank in this manner, one gives importance to the outgoing edges instead of the incoming edges.

## 2.4. Convolutional Neural Networks

In this section, we provide a surface-level explanation of *convolutional neural networks* and *transfer learning*. For more details, we refer to [38].

*Neural networks* are computational models inspired by the human brain, based on a set of neurons and the connections between them. The connections between these neurons emulate the synapse —the process in which a neuron sends a signal to a connected neuron. These networks are often defined by their *layers* (each layer containing a subset of neurons) and the connections between a layer and the previous one (Figure 2.5). The first layer is the input layer and the last layer is the output layer. The synapse process is emulated by adding the weighted input connections to the neuron and computing the image of this value by a non-linear function (*activation function*).

To compute the output of a neural network for an input, the activation for each neuron is computed, layer by layer. The performance of a neural network is evaluated by a loss function, which yields a measure of how incorrect the predictions of the neural network are. The lower the loss, the better. The objective is, then, to find the weights in the network's connections that minimize this loss. The process of tuning these weights to minimize loss is called *training* the neural network.

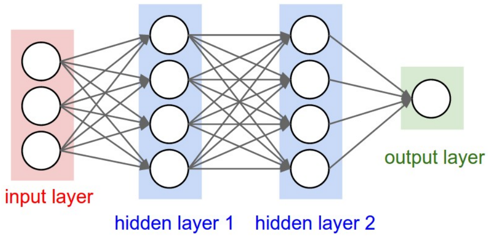



Figure 2.5: Diagram of a 3-layer neural network. Extracted from [38].

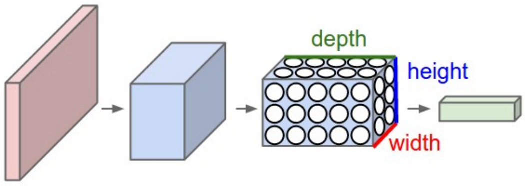



Figure 2.6: Diagram of a convolutional neural network. Extracted from [38].

The layers in traditional neural networks are defined as vectors of neurons with each layer fully connected to the previous one. Therefore, the input to a neural network can be any information that can be written in vector form. For instance, an image of width $W$, height $H$, and 3 channels (red, green, and blue) can be written as a vector of size $W \times H \times 3$. With this conversion a problem arises: we have lost the spatial information. To input images into the network and preserve this spatial information we have to adapt our neural networks. For this purpose, *Convolutional Neural Networks (CNNs)* were born. CNNs have the neurons in the layers arranged in 3 dimensions (Figure 2.6), and unlike conventional neural networks, the neurons in a layer will be only connected to a small area of the previous layer. This decreases the number of parameters of the network to make it feasible to work with images in the realm of hundreds of pixels wide and high, and makes it so a *convolution* operation takes place when the activation of a neuron is computed.

**The CRAFT text detector**. The Character Region Awareness For Text detection algorithm (CRAFT) is a text detector [1] based on a convolutional neural network, which, given an input image, enables us to get a set of polygons that contain the detected text in the image and a heat map of character detection (Figure 2.7). The main advantage of this implementation is that it is able to detect text of any shape with high accuracy.

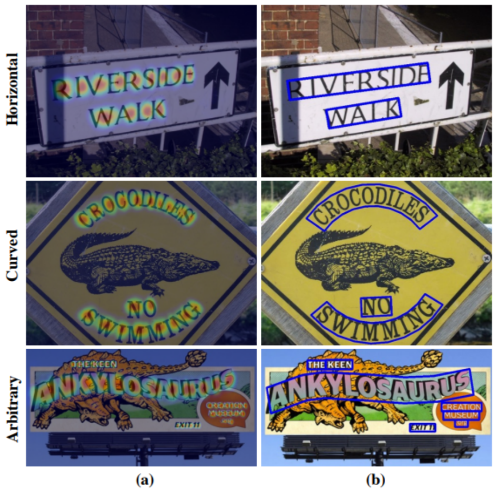



Figure 2.7: Results of the text detection using CRAFT. (a) Heatmaps. (b) Detection. Extracted from [1].

### 2.4.1. Transfer Learning

Training a neural network requires large amounts of labeled data. The process of collecting and labeling this data can be extremely time-consuming or even impossible if there is not much data related to the problem we are trying to solve. However, if our problem is a particular case of a problem that has already been attempted using a neural network with an identical structure to ours, we can leverage the weights learned in their training process and use them for our task. This method of choosing our network's weights is called *transfer learning*.

For example, there have been several attempts at classifying pictures into multiple categories depending on their content, such as the attempts at the ImageNet 1000 challenge, where a dataset of images and labels is provided and the goal is to classify each image in one of 1000 categories with the maximum accuracy possible. These categories are general enough that a neural network that performs well in this challenge effectively serves as a generic model of the visual world. We can then adapt this model to suit a more specific computer vision task by re-configuring the output layer of the model.

## 2.5. Inpainting

Inpainting is the process of improving the quality of an image by filling in parts of the image that were deteriorated. It can be applied to an image in any medium. The first modern use of inpainting can be traced to Pietro Edwards in the 18th century. Nowadays the process of inpainting an image can be performed digitally, manually with the help of specific software, or automatically using inpainting algorithms. The process of inpainting can be extended to be applied in areas that are not necessarily damaged, but that wish to be removed.

The basic idea of the inpainting algorithms is to replace selected areas with pixels similar to neighboring pixels so that the damaged area matches the region it is in. There have been several implementations of inpainting algorithms using different techniques. In this study, we use the Navier–Stokes inpainting algorithm [5], which is based on fluid dynamics. It sees the image intensity values as a stream for a two-dimensional incompressible flow. By solving the Navier–Stokes fluid dynamics equations for the desired areas, these areas will be filled with the surrounding fluid, which translates to being painted with the value of surrounding pixels.

# Chapter 3

# Related Work

Recent studies in network analysis have analyzed how culture and behavior are spread via social network ties, yet without focusing on the phenomenon that revolutionized culture spread in social networks, namely memes. Likewise, studies in computer vision have analyzed image memes and have attempted to detect memes or cluster them together, without taking a look at the users who post them. This study bridges the gap.

## 3.1. User Influence in Social Networks

Many studies have reported that behaviors or preferences of people can spread via social ties in social networks, mainly getting their knowledge through surveys [8, 11, 23]. However, to our knowledge, only [42] has derived characteristics of users from what the users post online.

In [42] a CNN was used to classify the images that a user posted on Facebook into 920 different categories that fall within sports, animals, clothes, food, furniture, music, plants, structures, places, scenes, and vehicles. For performing this task they trained the deep residual network ResNet-50 [18] with manually labeled data from the images that the users post online. Then, the categories of the images of the users were compared among friends and random users. Their results indicate that the content of the user's photographs is temporally correlated with that of the photographs of their friends, indicating that influence may occur. Additionally, as they also have geographic and temporal data for images, they could observe spatio-temporal trends like the dominance of Australia regarding surfing images and that of North and Central America regarding images containing tacos.

## 3.2. Meme Detection and Clustering

There have been studies that use memes and phrases extracted from news and blogs to track and study the dynamics of the news cycle [22] and research into clustering text-based memes on Twitter [15]. More in line with our study, the research in [44] was able to cluster image streams using perceptual image hashes (pHash). A perceptual image hash is an algorithm that produces a fingerprint of an image, with the property that visually similar images have similar fingerprints. They then assign the images using their fingerprints to clusters that were previously identified using meme annotation from sites such as "Know Your Meme".

One recent approach to meme detection is the Memesequencer model developed in [13]. In this study, they first separate the underlying image from a meme by matching the meme with a previously cataloged meme format, then they create an embedding using a variety of models. The best performing model was a multimodal deep learning approach, using a deep residual network ResNet-18 [18] for extracting image features concatenated with SkipThought text features. Finally, they construct an evolutionary tree for these meme formats. As noted, the research in [13] is limited to memes that have identifiable meme formats previously documented on sites like Memegenerator or Quickmeme.

Another approach to meme detection is the Meme-Hunter model from [6], which also uses multi-modal deep learning. Their model combines image features extracted with a ResNet-18 [18], text features extracted with a Long short-term memory neural network, and facial detection. This model is then used for detecting political memes on Twitter related to US and Swedish elections. Then, they map the evolution of the memes using nearest neighbors clustering with the features extracted from their detection model. Finally, they study how meme formats are used in left and right-wing contexts. Although the research in [6] shows great results, they only consider memes as pictures with superimposed text in impact font or text placed in white space over a picture. They do not give a reason for this criterion.

Contrary to these works, our approach to meme detection fits a much broader definition of meme and is more in line with the ever-changing landscape of memes, as it does not require the template to be previously cataloged.

# Chapter 4

# Formalization of the Problem

In this section, we detail concepts about memes and formalize the context of our problem by providing definitions that build on top of the concepts explained in Chapter 2. Through the study of a graph, we obtain a visual representation of the relationships between the users. Metrics derived from this graph serve as a tool to rank these users concerning their influence and get a deeper knowledge of the inner workings of meme virality in social networks.

**Definition 4.1.** A *meme* is a virally transmitted image embellished with text, usually sharing pointed commentary on cultural symbols, social ideas, or current events.

This definition of meme could be expanded to contain videos, text, or simply cultural references. However, within this study we only consider image memes. Examples of memes can be seen in Figure 4.1.


Figure 4.1: Three memes.

Figure 4.2: A meme format known as "Galaxy Brain" [28].

Given a meme, we refer to its *meme format* as the underlying image of the composition. A meme format can often be used to create more than one meme by adding or changing the existing embellishments. An example of a meme format is shown in Figure 4.2.

For any meme format, there exists a meme that used it first. Given a set of users $U$ and a meme format $F$, the *pioneer* of $F$ within $U$ is the user $u \in U$ who published the oldest meme with the format $F$. If the set of users is the set $U_{\text{tot}}$ of all users on all social media platforms, we refer to the pioneer as an *absolute pioneer*.

**Definition 4.2.** Given a set of users $U = \{u_0, \dots, u_n\}$, not necessarily belonging to the same social network platform, we define the *meme influence graph* of $U$ as a directed weighted graph $(U, E, w)$ with the following properties:

1. A pair $(u_i, u_j)$ with $i, j \in \{0, \dots, n\}$ and $i \neq j$ is in the set $E$ of edges if the user $u_j$ has posted a meme whose meme format was pioneered by $u_i$.
2. $w(u_i, u_j)$ is the number of memes posted by $u_j$ whose format was pioneered by $u_i$.

A meme influence graph $M = (U, E, w)$ is called *maximal* if $U = U_{\text{tot}}$, that is, if every user in every social media platform is in the set $U$.

The metrics depicted in Section 2.3 describe the following user characteristics when applied to a meme influence graph $(U, E, w)$:

- The *out-degree* of a user $u$ is the number of other users who have used a meme format pioneered by $u$.
- The *in-degree* of a user $u$ counts how many other users have pioneered meme formats that $u$ has used.
- The *weighted out-degree* of a user $u$ is the number of memes published by other users who have used a meme format pioneered by $u$.
- The *weighted in-degree* of a user $u$ indicates how many memes have been published by $u$ with a format pioneered by another user in $U$.

**Definition 4.3.** The *score* for a user $u$ is the value that the reverse PageRank algorithm (defined in Section 2.3.1) assigns to $u$.

For a maximal influence graph, degrees can be interpreted as follows. The out-degree of $u$ is the number of users who have been inspired by memes of $u$, while the in-degree is the number of users who have influenced $u$ when creating memes. The weighted out-degree of $u$ is the number of memes that have been influenced by $u$, while the weighted in-degree of $u$ is the number of memes from $u$ that have been influenced by some other user. Since a user, when creating a meme, can be inspired by a meme from a user who is not the pioneer of the meme format, the influence from a pioneer on a user is assumed to be indirect. In the case of a non-maximal influence graph $(U, E, w)$, we can also use the previous interpretations but with some nuances. Suppose that the pioneer is not the absolute pioneer of a meme format. In that case, there might not even be an indirect relationship of influence, since given a user $u_j$ in a set of users $U$ who published a meme with a format $F$ with pioneer $u_i \in U$, there exists a possibility that $u_j$ first saw the format $F$ from another user $u_l \notin U$. Therefore, the relationship of influence on a meme influence graph that is not maximal has to be interpreted as potential influence.

# Chapter 5

# Implementation

The process for building a meme influence graph (Definition 4.2) is outlined in Figure 5.1. The input to this system is a set of users $U$ and the output is the meme influence graph for that set of users. In this chapter we describe a detailed implementation of this system.

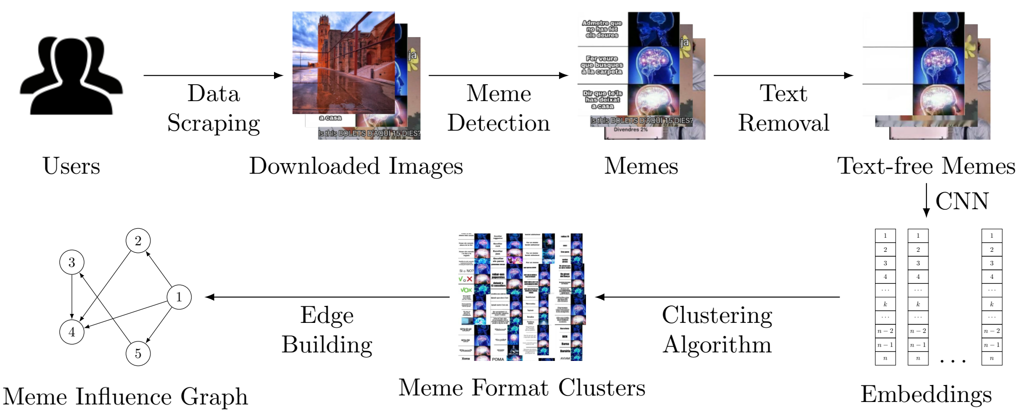



Figure 5.1: Flow diagram for the creation of a meme influence graph.

## 5.1. Data Scraping

Even though a meme influence graph is defined for any users belonging to any social media platform, in this project we have limited the implementation's scope to one social media platform. The chosen social media platform is Instagram, for the reasons outlined in Section 2.1. Data for Instagram were extracted by storing the responses of Instagram's API to the calls that the browser made when browsing relevant data. The data accessed in this study are 100% public and accessible by anyone.

Table 5.1: Features stored from each of the users. Features under the dashed line are computed after processing all the posts from the user.

| **Feature** | **Type** | **Description** |
|---|---|---|
| username | String | Username of the user |
| full_name | String | Name of the user |
| biography | String | Description of the user profile |
| is_business | Boolean | Whether the user has a business account |
| is_private | Boolean | Whether the user profile is private or public |
| is_verified | Boolean | Whether the user profile |
| follower_count | Integer | Number of users following this user |
| following_count | Integer | Number of users this user follows |
| post_count | Integer | Number of posts published by this user |
| profile_pic_url | String | URL to the user's profile picture |
| average_comments | Float | Average number of comments per post |
| average_likes | Float | Average number of likes per post |
| average_views | Float | Average number of views per post |
| meme_ratio | Float | Percentage of posts that have been detected as memes by the meme detection |

Table 5.2: Features stored from the publications. Features under the dashed line are computed after processing the image. Features with an asterisk are only valid for videos.

| **Feature** | **Type** | **Description** |
|---|---|---|
| caption_text | String | Text of the post |
| code | String | Text that identifies the URL |
| is_video | Boolean | Whether the media is a video |
| comment_count | Integer | Number of comments in the post |
| view_count* | Integer | Number of views of the post |
| like_count | Integer | Number of likes on the post |
| taken_at | Timestamp | Date and time the post was published in |
| thumbnail_url | String | URL to the thumbnail of the media |
| video_url* | String | URL to the video |
| user.username | String | Username of the user that published the post |
| user.full_name | String | Full name of the user that published the post |
| usertags | Array of Strings | Hashtags associated with the post |
| is_meme | Boolean | Whether the image has been detected as a meme by the meme detection |

Retrieving user data is not strictly necessary for building a meme influence graph, but having some user characteristics enables us to interpret the graph and compare the metrics from the graph to the existing features of the users. In Table 5.1 we can see a list of the relevant features from the users extracted from the platform.

When browsing the user's publications (or posts), the only essential information is the images within them. As with the user data, features from publications were also valuable for later study, so they were extracted as well. Features extracted from posts can be seen in Table 5.2. In this process, videos have been treated as images using their first frame. The first frame of a video is a good representation of the media in this context, since meme videos using the same format have very similar first frames.

## 5.2. Meme Detection

**Algorithm 1** Meme Detection Algorithm

**Require:**
$I$ := image to process of size $(w, h)$
$\alpha, \beta$ := lower and upper bounds for the text-to-image area ratio
$\gamma$ := minimum standard deviation threshold
**Ensure:**
*true* if the image is detected as a meme, *false* otherwise

$p \leftarrow$ text detection$(I)$ ▷ Detect areas containing text on the image using CRAFT [1] text detector
$A_{\text{tot}} \leftarrow w \times h,\ A_{\text{text}} \leftarrow \text{area}(p)$ ▷ Compute total image area and text area
$r \leftarrow A_{\text{text}} / A_{\text{tot}}$ ▷ Compute text-to-image area ratio
**if** $r \notin (\alpha, \beta)$ **then** ▷ If the text-to-image area ratio is not within bounds
  **return** *false* ▷ The image has either only text or no text and it is not a meme
**end if**
$I_{\text{inpainted}} = \text{inpaint}(I, p)$ ▷ Inpaint $I$ using Navier–Stokes [5] with $p$ as inpainting mask
$\sigma = \text{std}(I_{\text{inpainted}})$ ▷ Compute standard deviation of grayscale values of the inpainted image
**if** $\sigma \leq \gamma$ **then** ▷ If after text removal the image has high grayscale deviation
  **return** *false* ▷ There was no content of substance left after removing the text, so it is not a meme
**else** ▷ If after text removal the image has low grayscale deviation
  **return** *true* ▷ There was an underlying image after removing the text, so the image is a meme
**end if**

For detecting which of the downloaded images are memes, none of the methods previously used (detailed in Chapter 3.2) were useful in this context, since they are extremely limited in their definition of meme, so a new algorithm was required. In line with our broad definition of meme, the task that the meme detection algorithm had to perform was to discard images with no text or no underlying image. The process used to perform this task is described in Algorithm 1.

Our meme detection algorithm starts by detecting the regions containing text in the image with the CRAFT text detector (Section 2.4). From the text regions, we compute the total text area by using Gauss's area formula. Then, we compute the total image area and the text-to-image area ratio $r$. To discard images that have no sign text or are mostly text, if $r$ is not within our upper and lower bounds of text-to-image area ratio (denoted by $\alpha$ and $\beta$, respectively), we conclude that the image is not a meme.

Now we are onto the task of detecting images that have no underlying image in their composition. This was not accomplished by the previous step because text-only images can contain white space around the text for aesthetic purposes (padding, decoration...) and that drives the text-to-image area ratio down. To determine if the area not containing text is of substance, we first remove the text detected in the previous set using the Navier–Stokes inpainting algorithm (Section 2.5) with the detected text areas as the inpainting mask. Once the text has been removed from the image, we evaluate if there is any content left by computing the standard deviation $\sigma$ of the gray level values of the inpainted image converted to grayscale. The value of $\sigma$ can be thought of as a measure for image contrast. If $\sigma$ is lower than a specified threshold $\gamma$, we conclude that the image is not a meme, while if it is higher then the input image is classified as a meme.

## 5.3. Embeddings

From a text-free meme, we have to extract features to have a lower-dimensional representation of the source image that enables us to determine differences in content between two images by comparing their features. Using text-free memes instead of the original memes with text makes the underlying meme format exposed. This diminishes the differences between memes using the same meme format and makes it easier to cluster them together in the next step.

We use the convolutional neural network VGG16 [37] for extracting these features. We leveraged transfer learning (Section 2.4.1) by using the network with the weights from the ImageNet challenge. In the context of the ImageNet challenge, the network had to classify images into 1000 different categories. To adapt the network to the task at hand, we set the output to the second-to-last fully connected layer, bypassing the classifier layers and giving an output of 4096 dimensions (Figure 5.2). This neural network and weights combination was chosen because it gave good results for characterizing memes in the past in [2] which had a broader meme definition than [6] and [13].

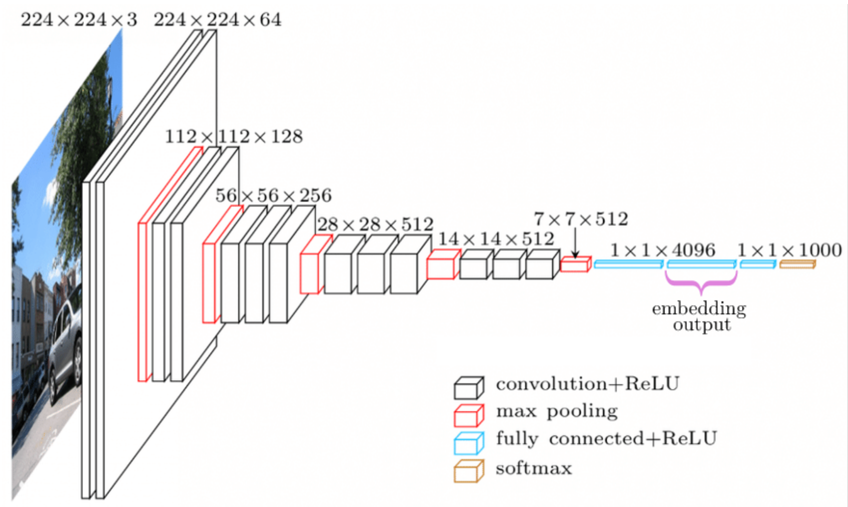



Figure 5.2: *VGG-16*'s architecture, with the output for our embeddings in pink. Adapted from [25].

## 5.4. Deep Image Clustering

To cluster memes into groups sharing the same meme format, we input the embeddings into a clustering algorithm, namely DBSCAN (Section 2.2.1). We apply Principal Component Analysis (PCA) to reduce the dimensionality of the samples to 1024 for improving efficiency. The main downside to this algorithm is that it is not an incremental algorithm and every time new samples are added, the algorithm has to recompute the cluster label for the entirety of the samples.

## 5.5. Meme Influence Graph

Finally, we build the meme influence graph (Definition 4.2). We add input users as nodes and then, for each cluster, we create edges from the pioneer of a cluster to the authors of the rest of the memes of that cluster (Algorithm 2). After building the graph, we compute the metrics defined in Section 4.

**Algorithm 2** Meme Influence Graph Construction

**Require:**
    $U :=$ a set of users
    $C :=$ the set of clusters containing the memes of the users in $U$
**Ensure:**
    $(N, E, w)$ a meme influence graph for $U$

```
N ← U
for all c ∈ C do
    p ← earliest meme in c
    for all m ∈ c \ {p} do
        if p.user ≠ m.user then
            w((p.user, m.user)) += 1
            if (p.user, m.user) ∉ E then
                add (p.user, m.user) to E
            end if
        end if
    end for
end for
```

# Chapter 6

# Results

This chapter contains the results of using our implementation to build the meme influence graph for a selected set of users. This implementation has been coded using Python, and a NoSQL database (MongoDB) has been used for locally storing the data generated at each step of the process.

## 6.1. Data Scraping

The users were selected starting from a small set of users who publish memes and adding more users to the set by exploring the recommendations of similar users that Instagram offers. The initial set of users contained accounts that publish memes in Spanish and Catalan. This criterion was established to obtain a set of users that we expect to be densely connected in their meme influence graph. The time frame of the posts was limited to a period comprised between January 1st, 2017 and April 23rd, 2022. In this set, there are 91 users and 457,101 media. Within it, we find users who post general topic memes but also some who post topic-specific memes, such as football-themed, music-themed, or region-themed.

## 6.2. Meme Detection

The meme detection parameters $(\alpha, \beta, \gamma)$ were found experimentally by selecting a random sample of the images on the dataset and splitting it into memes and non-memes using the meme detection algorithm. False positives and false negatives were manually identified and the thresholds were adjusted. This step was repeated several times until adjustments to the values were negligible. With this procedure, the following values were found: $\alpha = 0.018$, $\beta = 0.4$, $\gamma = 26$. Since standard deviation can vary depending on the size of the image, the images were resized to 224 × 224 pixels, matching the required dimensions for an input image to the neural network VGG16. Through our findings, we notice that JPEG compression can introduce artifacts that artificially raise the standard deviation of the image. Hence images that have been compressed several times show a higher standard deviation.

In Figs. 6.1 and 6.2 we can see examples of how our meme detection algorithm processes the images. After applying the meme detection algorithm to all the images in our dataset, 342,984 out of 457,101 images (75%) were detected as memes.

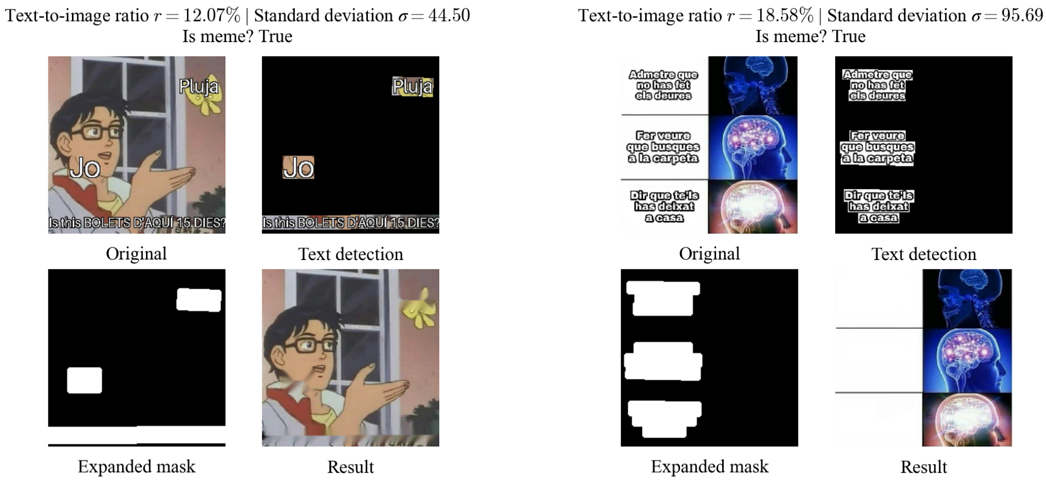



Figure 6.1: Images correctly detected as memes. Classification is based on text-to-image area ratio and standard deviation of gray values of the inpainted image converted to grayscale (Algorithm 1).

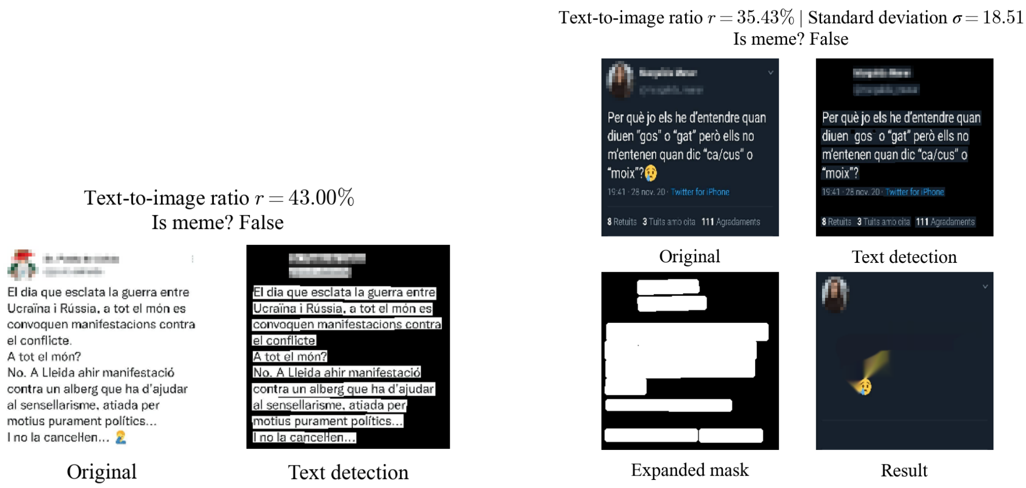



Figure 6.2: Images correctly classified as non-memes because of high text area and low standard deviation of grayscale values, respectively (Algorithm 1).

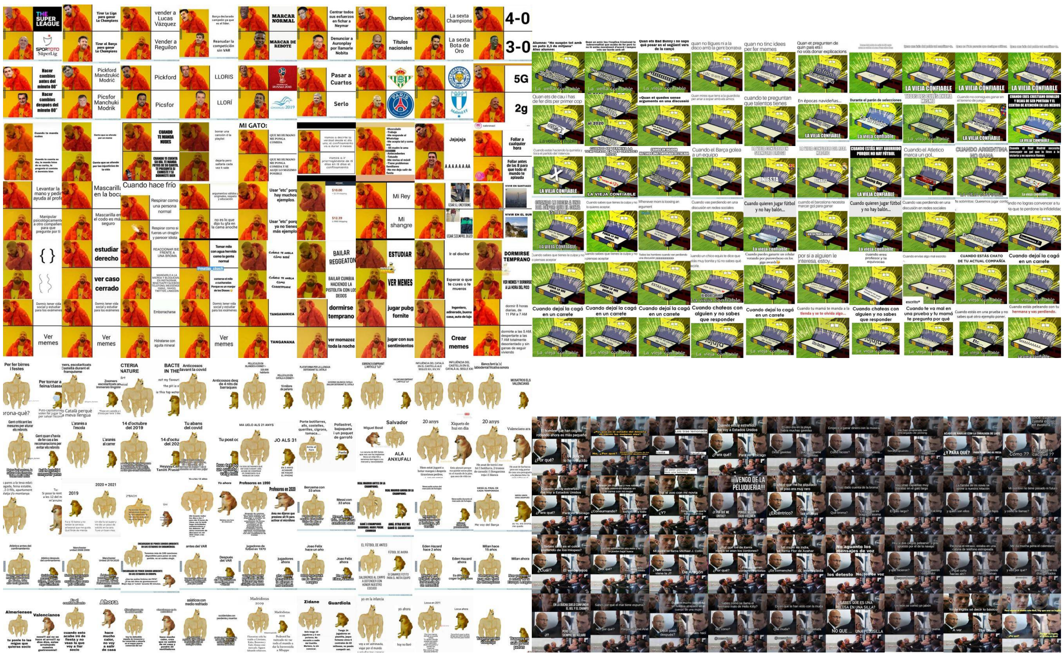


Figure 6.3: Some of the images in four clusters.

## 6.3. Embeddings and Clustering

The inputs to the VGG16 neural network were the text-free memes generated in the previous step, in a size of $224 \times 224$ pixels and preprocessed according to the needs of the neural network: the images were converted from RGB to BGR; then each color channel was zero-centered with respect to the ImageNet dataset. The embeddings were then reduced in dimensionality to 1024 components using PCA. By using 1024 components, we retain 87.65% of the variance and the time spent for clustering went from 7 hours 44 minutes to 1 hour and 8 minutes.

The DBSCAN clustering algorithm was used with the cosine distance as a metric, a minimum samples per cluster value of 3, and an epsilon value $\epsilon = 0.12$. The epsilon parameter for the algorithm was tuned manually by selecting a small number of memes with popular meme formats and visualizing their clusters with an initially big epsilon value. The epsilon value was lowered in small increments until the only memes left in the selected meme's cluster were memes with the same meme format. The clustering was able to group memes using the same meme format (Figure 6.3). On our dataset, the algorithm found 13,663 clusters containing 82,801 memes, and 260,183 memes were detected as noise.

Here we notice the main downside of using transfer learning instead of re-training the network to our specific problem. The embeddings (and by extension cluster labels) show a clear bias towards ImageNet categories, namely if a meme fits well in a certain ImageNet category it is very likely that it will be clustered with images in the same category. For example, all memes featuring dogs of the breed *Golden Retriever* are clustered together even though they might not use the same meme format (Figure 6.4). Lowering the epsilon value for the clustering algorithm does not fix this issue since memes that fall in the same ImageNet category are closer in our feature vector space than memes that share the same meme format but do not fall in a specific category. The clusters that were too broad and incorporated multiple meme formats were discarded when building the graph.

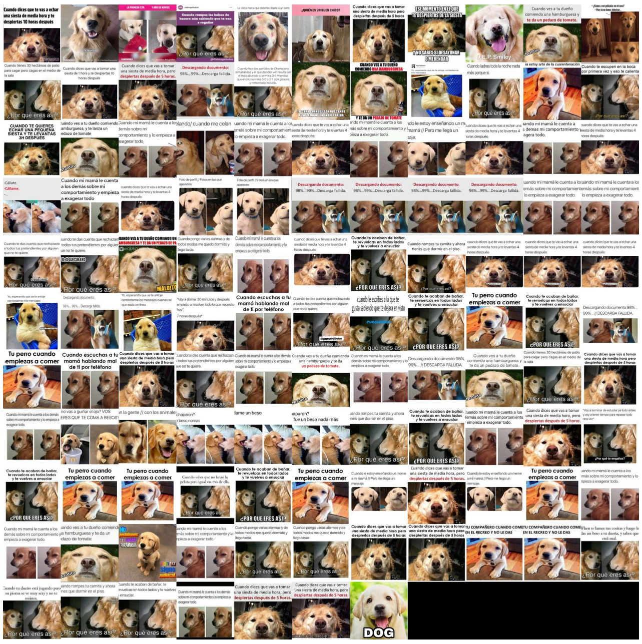


Figure 6.4: Cluster of memes featuring dogs of the breed *Golden Retriever*, category #207 of the ImageNet challenge. It can be appreciated that most of the memes do not share the same meme format.

In this step, we can also observe images that were false positives in our meme detection algorithm. Since these false positives defy our meme detection for similar reasons, they are often clustered together and can be removed from our data as a group. In Figure 6.5 we can see clusters with false positives. These images are images that after text removal still have high contrast. Therefore, their standard deviation was higher than our threshold and were wrongly classified as memes.

Another challenge was dealing with the watermarks that users add to the memes when creating their own versions of memes. Most of these watermarks are just a superposition of their username, and since it is text, it gets removed by

Figure 6.5: Clusters from the main dataset that were discarded after clustering because the images were false positives in the meme detection. The cluster on the left contains tweets with colored backgrounds and the one on the right screenshots from text messaging apps.

the meme detection step. However, we observed that some users use a logo- or symbol-based watermark instead of a text-based one, which cannot be removed by our processing. This makes it so the memes from this user will be further apart from the rest of the memes sharing the same meme template and, in extreme cases, might form their own cluster.

## 6.4. Influence Graph and Metrics

We built the meme influence graph and computed the metrics defined in Section 4. The graph for our anonymized set of users is shown in Fig. 6.6, where high score nodes can be easily identified. By computing the Pearson correlation coefficient between each of the pre-existing characteristics of the users and each of our metrics, we found that the follower count had low positive correlations with score ($\rho = 0.29$), weighted in-degree ($\rho = 0.27$), and weighted out-degree ($\rho = 0.30$). Hence we conclude that the number of followers, which is frequently used for determining the importance of a user [19], was unable to tell the difference between incoming influence and outgoing influence in our set of users. The number of posts published by these users had a high correlation with score ($\rho = 0.69$), weighted in-degree ($\rho = 0.85$), and weighted out-degree ($\rho = 0.73$). This matches the intuition that the more posts a user makes, the more opportunities for their

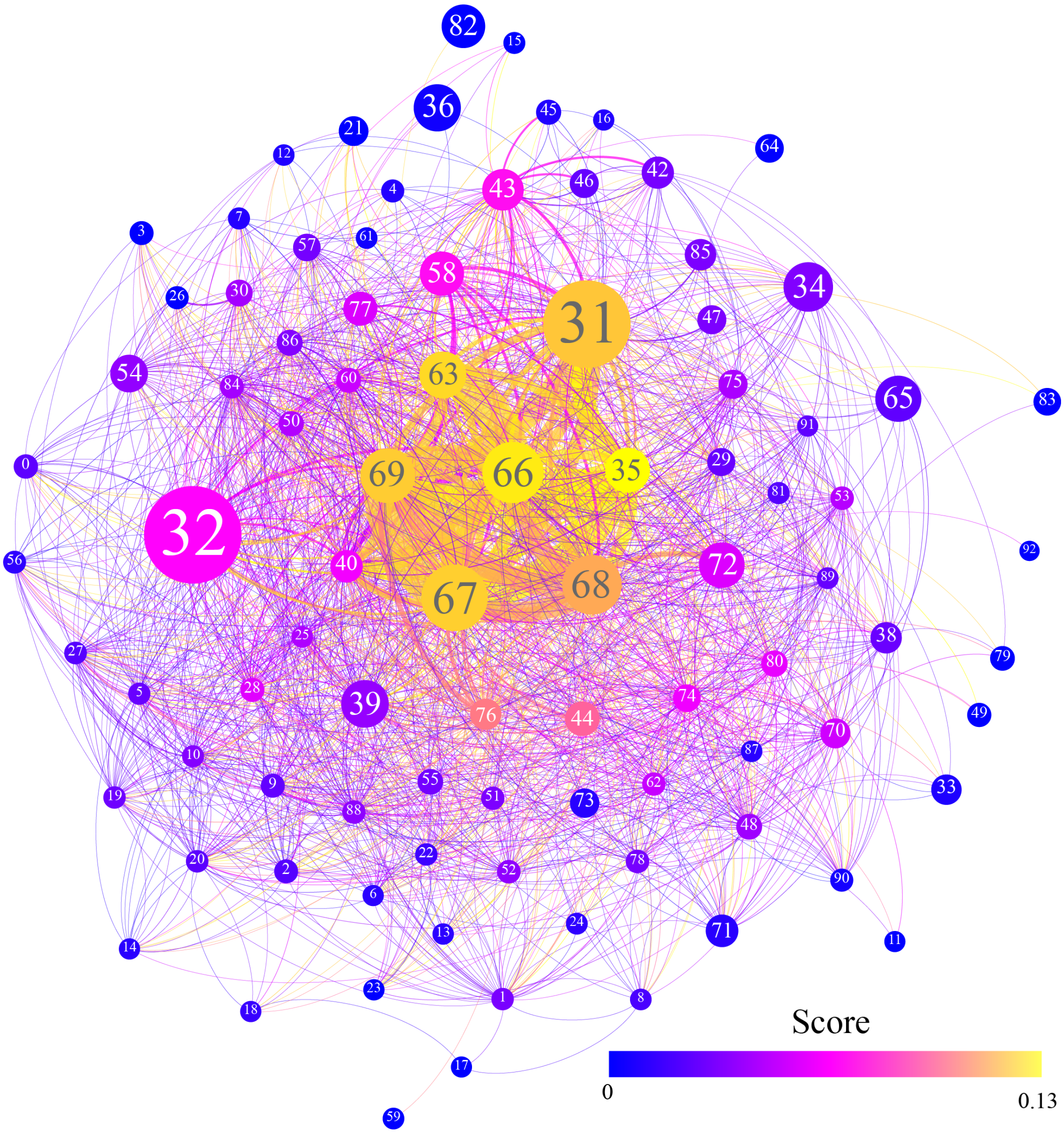



Figure 6.6: The meme influence graph for our set of 91 users. Nodes are labeled from 1 to 91; node size represents follower count; node color represents score from 0 (blue) to 0.13 (yellow); edge thickness represents edge weight; edge color matches the source node color; edge directions are represented clockwise.

memes to influence or be influenced. No correlation higher than 0.10 was found for average likes and comments per post with influence per post (weighted out-degree/media count), indicating that user engagement in posts does not correlate significantly with more influential memes. Meme-to-image ratio also had no significant correlation with our metrics, showing that how dedicated a user is to posting memes does not correlate to how influential his memes are. Further correlation coefficient values between metrics and user characteristics are shown in Figure 6.7.

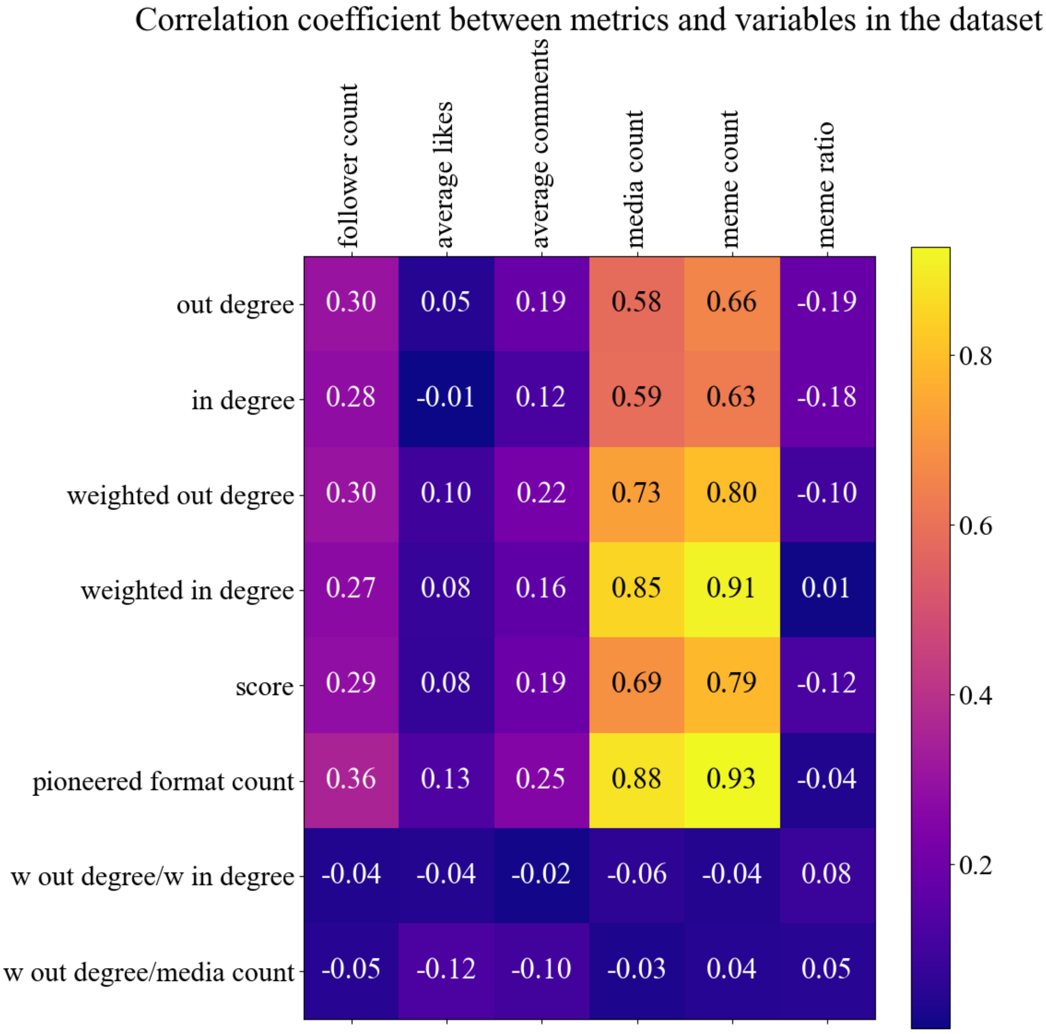
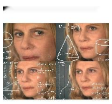



Figure 6.7: Correlations between graph metrics ($y$ axis) and user features ($x$ axis). The color scale denotes absolute value.

Although most meme formats in this dataset were not topic- or region-specific, we found communities of users sharing memes that match a certain topic or geographic area using the Clauset–Newman–Moore [12] community detection algorithm on our graph. There were communities posting football-related memes and others related to territories. Most users were not included in any community with a relevant trait.

Table 6.1: Meme formats of the main dataset ranked by the number of unique users that have used the format.

| # | | Meme format | #users (%) | #memes (%) |
|---|---|---|---|---|
| 1 |  | *Yes Chad* [35] | 45 (49.45%) | 191 (0.05%) |
| 2 |  | *Drakeposting* [27] | 34 (37.36%) | 238 (0.07%) |
| 3 |  | *Galaxy Brain* [28] | 33 (36.26%) | 143 (0.04%) |
| 4 |  | *Swole Doge vs. Cheems* [31] | 30 (32.96%) | 85 (0.02%) |
| 5 |  | *Types of Headaches* [33] | 26 (28.57%) | 70 (0.02%) |
| 6 |  | *Woman Yelling at a Cat* [34] | 25 (27.47%) | 76 (0.02%) |
| 7 |  | *Math Lady / Confused Lady* [30] | 25 (27.47%) | 56 (0.02%) |
| 8 |  | *Awkward Look Monkey Puppet* [26] | 24 (26.37%) | 78 (0.02%) |
| 9 |  | *I Bet He's Thinking About Other Women* [29] | 24 (26.37%) | 78 (0.02%) |
| 10 |  | *Traumatized Mr. Incredible* [32] | 24 (26.37%) | 45 (0.01%) |

## 6.5. Meme Formats

By processing all the necessary data for building the meme influence graph, we not only have an insight into the users in our dataset but also about the meme formats themselves. We now have data from which we can extract their popularity (Table 6.1) and temporal use (Figure 6.8). The most popular formats in our dataset are popular enough on the global scale that they have been cataloged on sites like “Know Your Meme” and they have an assigned name. By using Google Trends (`trends.google.com`) we can get a measure of the popularity of a search term in the Google search engine in a particular time frame. We extracted this data for the names of the 5 most popular meme formats in our dataset in the timeframe that our images were posted in. We then superposed our data and the data from Google (Figure 6.9). We can observe that the use of the meme formats by our Spanish creators generally matches the popularity of the names of the meme formats as search terms, except in the case of the *Galaxy Brain* format. We can also notice that our users were months late to the popularity of the *Yes Chad* format, and early to the popularity of the *Woman Yelling at a Cat* format.

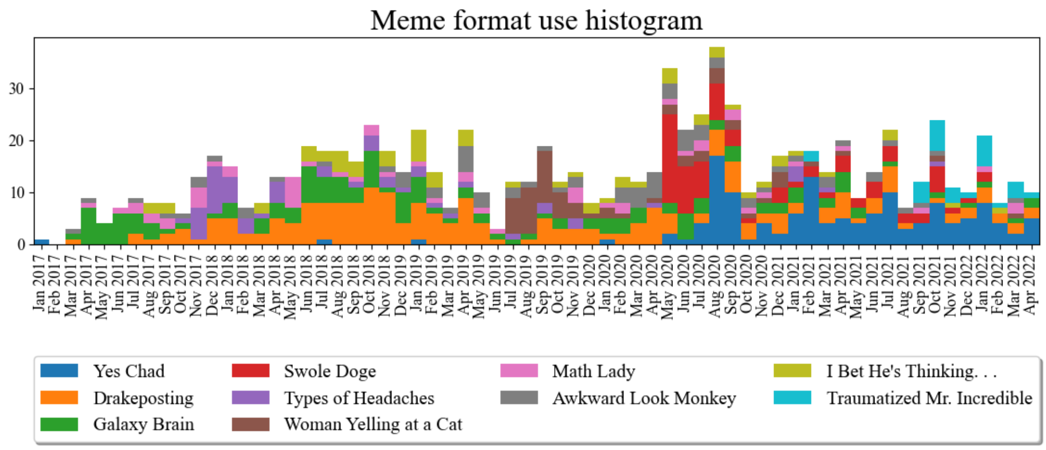



Figure 6.8: Histogram showing the number of memes published using a given meme format each month.

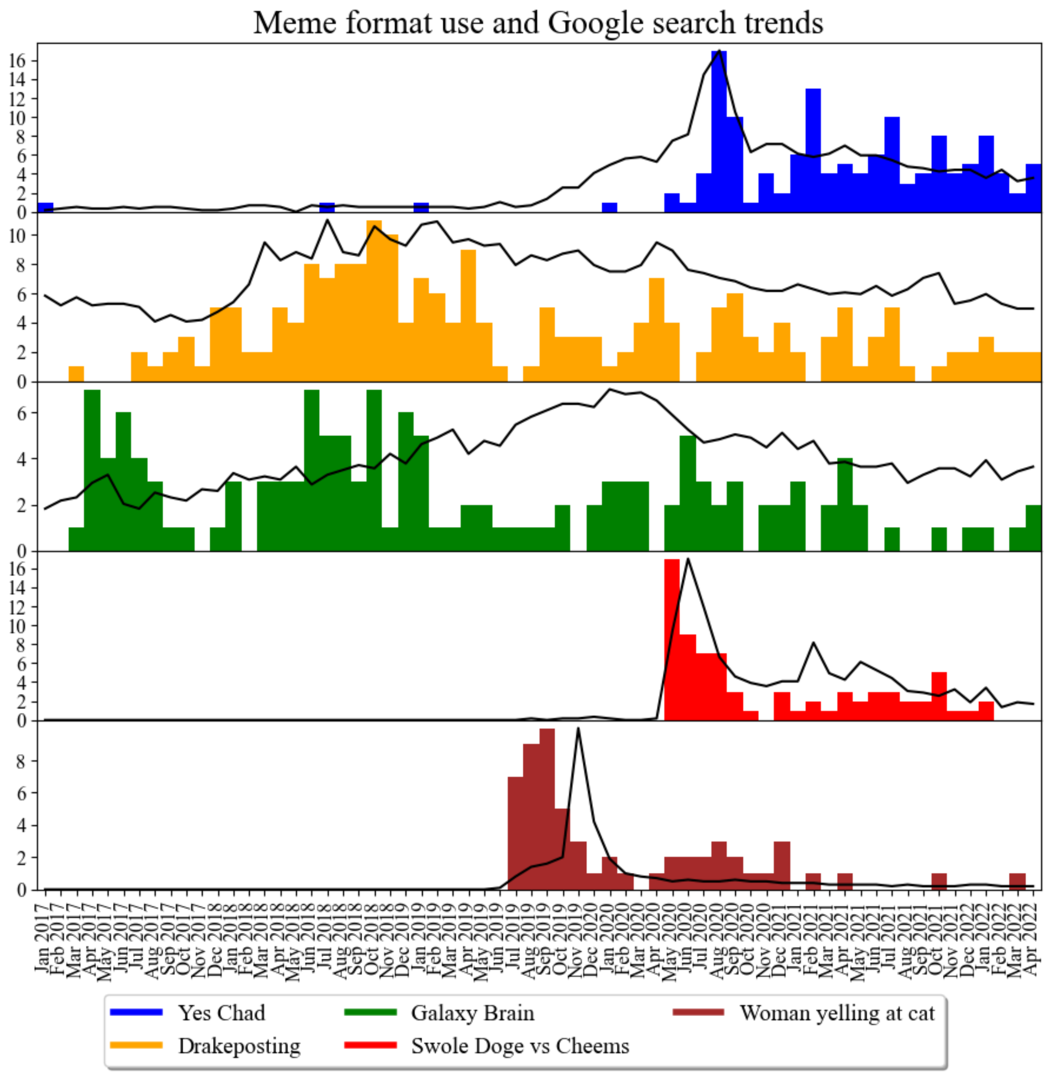



Figure 6.9: Comparison of the top 5 meme formats use each month (box plot) with the google trends data for searching the name of the meme template (line plot). The search terms match the meme name except in the case of *Drakeposting*, where *Drake meme* was used as a search term since the term *Drakeposting*'s popularity was very low. The *types of headaches* meme format was discarded because the name has multiple meanings as a search term.

# Chapter 7

# Conclusions and Future Work

We have presented a graph and metrics that serve as tools to visualize the influence and the relationships of meme creators, and provided a pipeline for constructing the graph and computing the metrics. Our objectives have been achieved by building a pipeline for extracting and storing large amounts of data from Instagram, implementing our meme detection algorithm, extracting features from image data using a CNN, and clustering images based on their underlying images using DBSCAN. By applying our definitions to a set of users who post memes we have verified our hypothesis, as by building the graph and computing the metrics we have been able to visualize relationships between users, rank them, and show that some of these users would be overlooked by using standard metrics for determining influence. Finally, we have also been able to give an insight into the memes that these users post.

Our ranking method can be applied, for example, to select candidates from a set of users for a marketing campaign using memes. Since users with high scores are those whose content has been influenced the least by other users in the graph and are the most likely to have influenced other users, by basing our criteria on their scores we ensure that memes generated by the selected users have the highest chance of viral spread through other users and reach an audience bigger than their group of initial followers. Using our graph, we can detect users who can be considered as “hidden gems”, that is, users with a high score although they may not rank high concerning their number of followers. For example, user #35 in Fig. 6.6 has the highest score but ranks 13th regarding the number of followers. By finding these “hidden gems”, we can optimize the budget of a campaign by selecting users with a high score-to-follower ratio.

## 7.1. Limitations

The purpose of the data used for evaluation in this work is to illustrate how our graph and metrics can be applied. This small-scale experiment does not attempt to characterize Instagram as a social network or extract information about general meme format use or virality, although observing such characteristics is not out of the question if the experiment is scaled to encompass a significant number of users. The set of users chosen for this example does not represent a general population of users; therefore, the methodology can be extrapolated to other sets of users but not the results. The metrics and connections also need to be carefully interpreted according to their definition and interpretation as explained in Section 4, since it is very likely that users have relationships with other users who are not represented in the graph.

## 7.2. Future Work

Future studies can expand on the foundations that this project lays by different means. First, the feature extraction model could be improved by training it from scratch or by concatenating it with other models such as face detection, text sentiment analysis, or text feature extraction. This later improvement would transform the model into a multi-modal model, following the steps of the related work in [6, 44]. Second, the implementation could be made more scalable by using an incremental clustering algorithm such as the incremental DBSCAN clustering algorithm provided in [9], which would not require computing the cluster labels for all the samples when a new one is added. Finally, our methodology could be implemented on a representative sample of users for a social network, region, or topic, from which significant conclusions regarding influence and meme format use could be derived.

# Bibliography


[1] Youngmin Baek et al. "Character region awareness for text detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9365–9374.

[2] Kate Barnes et al. "Dank or not? Analyzing and predicting the popularity of memes on Reddit". In: *Applied Network Science* 6.1 (2021), pp. 1–24.

[3] Christian Bauckhage. "Insights into internet memes". In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 5. 1. 2011, pp. 42–49.

[4] Christian Bauckhage, Kristian Kersting, and Fabian Hadiji. "Mathematical models of fads explain the temporal dynamics of internet memes". In: *AAAI Conference on Web and Social Media*. Vol. 7. 1. 2013, pp. 22–30.

[5] Marcelo Bertalmio, Andrea L Bertozzi, and Guillermo Sapiro. "Navier-stokes, fluid dynamics, and image and video inpainting". In: *2001 IEEE Computer Society Conference on CVPR*. Vol. 1. IEEE. 2001, pp. I–I.

[6] David M Beskow, Sumeet Kumar, and Kathleen M Carley. "The evolution of political memes: Detecting and characterizing internet memes with multimodal deep learning". In: *Information Processing & Management* 57.2 (2020), p. 102170.

[7] Susan Blackmore et al. "The power of memes". In: *Scientific American* 283.4 (2000), pp. 64–73.

[8] Robert M Bond et al. "A 61-million-person experiment in social influence and political mobilization". In: *Nature* 489.7415 (2012), pp. 295–298.

[9] Sanjay Chakraborty and Naresh Kumar Nagwani. "Analysis and study of Incremental DBSCAN clustering algorithm". In: *arXiv preprint arXiv:1406.4754* (2014).

[10] Jenn Chen. *Instagram statistics you need to know for 2022*. 2022. URL: `https://web.archive.org/web/20220513234913/https://sproutsocial.com/insights/instagram-stats/`.

[11] Nicholas A Christakis and James H Fowler. "Social contagion theory: examining dynamic social networks and human behavior". In: *Statistics in medicine* 32.4 (2013), pp. 556–577.

[12] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. "Finding community structure in very large networks". In: *Physical review E* 70.6 (2004), p. 066111.

[13] Abhimanyu Dubey et al. "Memesequencer: Sparse matching for embedding image macros". In: *Proceedings of the 2018 World Wide Web Conference*. 2018, pp. 1225–1235.

[14] Martin Ester et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *kdd*. Vol. 96. 34. 1996, pp. 226–231.

[15] Emilio Ferrara et al. "Clustering memes in social media". In: *Advances in Social Networks Analysis and Mining*. IEEE. 2013, pp. 548–555.

[16] Google. *Clustering Algorithms*. 2020. URL: `https://web.archive.org/web/20210807181253/https://developers.google.com/machine-learning/clustering/clustering-algorithms`.

[17] Markus Götz, Christian Bodenstein, and Morris Riedel. "HPDBSCAN: highly parallel DBSCAN". In: Nov. 2015, pp. 1–10. DOI: `10.1145/2834892.2834894`.

[18] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[19] *Influencer Marketing: A Research Guide*. 2020. URL: `https://web.archive.org/web/20210627064440/https://guides.loc.gov/influencer-marketing/metrics-and-costs`.

[20] Keepios. *Global social media statistics*. 2022. URL: `https://web.archive.org/web/20220513111601/https://datareportal.com/social-media-users`.

[21] Liisi Laineste and Anastasiya Fiadotava. "Globalisation and ethnic jokes: A new look on an old tradition in Belarus and Estonia". In: *The European Journal of Humour Research* 5.4 (2017).

[22] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. "Meme-tracking and the dynamics of the news cycle". In: *15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2009, pp. 497–506.

[23] Kevin Lewis et al. "Tastes, ties, and time: A new social network dataset using Facebook. com". In: *Social networks* 30.4 (2008), pp. 330–342.

[24] Chen Ling et al. "Dissecting the meme magic: Understanding indicators of virality in image memes". In: *Proceedings of the ACM on Human-Computer Interaction*. Vol. 5. CSCW1. ACM New York, NY, USA, 2021, pp. 1–24.

[25] Manolis Loukadakis, José Cano, and Michael O'Boyle. "Accelerating Deep Neural Networks on Low Power Heterogeneous Architectures". In: Jan. 2018.

[26] Know Your Meme. *"Awkward look monkey puppet" Meme Format*. 2022. URL: https://knowyourmeme.com/memes/awkward-look-monkey-puppet.

[27] Know Your Meme. *"Drakeposting" Meme Format*. 2022. URL: https://knowyourmeme.com/memes/drakeposting.

[28] Know Your Meme. *"Galaxy Brain" Meme Format*. 2022. URL: https://knowyourmeme.com/memes/galaxybrain.

[29] Know Your Meme. *"I bet he's thinking about other women" Meme Format*. 2022. URL: https://knowyourmeme.com/memes/bet-hes-thinking-about-other-women.

[30] Know Your Meme. *"Math lady" Meme Format*. 2022. URL: https://knowyourmeme.com/memes/math-lady-confused-lady.

[31] Know Your Meme. *"Swole Dog vs Cheems" Meme Format*. 2022. URL: https://knowyourmeme.com/memes/wole-doge-vs-cheems.

[32] Know Your Meme. *"Traumatized Mr Incredible" Meme Format*. 2022. URL: https://knowyourmeme.com/memes/raumatized-mr-incredible-people-who-dont-know-vs-%20people-who-know.

[33] Know Your Meme. *"Types of Headaches" Meme Format*. 2022. URL: https://knowyourmeme.com/memes/types-of-headaches.

[34] Know Your Meme. *"Woman yelling at cat" Meme Format*. 2022. URL: https://knowyourmeme.com/memes/oman-yelling-at-a-cat.

[35] Know Your Meme. *"Yes Chad" Meme Format*. 2022. URL: https://knowyourmeme.com/memes/yes-chad.

[36] Lawrence Page et al. *The PageRank citation ranking: Bringing order to the web.* Tech. rep. Stanford InfoLab, 1999.

[37] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[38] Stanford. *Convolutional Neural Networks for Visual Recognition*. 2022. URL: https://cs231n.github.io/.

[39] Lilian Weng et al. "Competition among memes in a world with limited attention". In: *Scientific Reports* 2.1 (2012), pp. 1–9.

[40] Tony Wibowo et al. "Usage Of Meme As Information Sharing Media". In: *JURNAL ILMIAH BETRIK: Besemah Teknologi Informasi dan Komputer* 11.3 (2020), pp. 165–171.

[41] Dongkuan Xu and Yingjie Tian. "A comprehensive survey of clustering algorithms". In: *Annals of Data Science* 2.2 (2015), pp. 165–193.

[42] Quanzeng You et al. "Cultural diffusion and trends in facebook photographs". In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 11. 1. 2017, pp. 347–356.

[43] Burcu Yucesoy and Albert-László Barabási. "Untangling performance from success". In: *EPJ Data Science* 5.1 (2016), pp. 1–10.

[44] Savvas Zannettou et al. "On the origins of memes by means of fringe web communities". In: *Internet Measurement Conference*. 2018, pp. 188–202.