Final Project

Report

BIU CS – Final Project, Year 3.

Students:

## Omri Fridental

## Guy Wolf

## Oren Schwartz

Instructors:

Dr. Ariel Roth

Asi Barak

Course:

סדנה לפרויקטים – 89385

# 01 Our vision

Our vision was based on a real-life problem – gallery management. Gallery management is important for many different reasons (memory management, quick search, easy access and more). We wanted to explore a subproblem of gallery management which we call “AI filtering”.

The basic idea behind “AI filtering” is that some users sometimes want to filter their gallery based on some non-trivial mathematical function. This function is often hard to define mathematically. Therefore, we use a machine learning algorithm to train on datasets that have been pre-classified, in hope that our model can estimate the unknown function with little to no mistakes[[1]](#footnote-1).

With the main idea in mind, Omri has suggested that we explore a specific unknown function so that we can create an application that can be used by users who commonly search those pictures in their gallery. We have decided to choose a complicated yet useful filter function – Memes.

Why did we choose to filter by memes?

We chose to filter by memes because of 3 main reasons:

1. Our personal galleries are filled with hundred or even thousands of pictures that can be classified as memes.

2. We thought this is a unique idea for a classifier, as far as we know there is not an easy tool that can accomplish this specific task.

3. The choice of this topic has led us to another very interesting problem, creating another model that takes our main model’s results and classify the memes into groups aka finding the “pattern” of the meme.

A visual Explanation:

TODO: explain what a meme is, explain the idea behind a pattern.

# 02 Models

As we previously mentioned our application features 2 different models –

1. A model that classifies whether a specific image is a meme (aka the “Meme Classifier”).

2. A model that finds a given meme its pattern (aka the “Perceptual Hash”).

How do the Meme Classifier work?

TODO: fill the training process, fill the model’s final specs

How do the Perceptual Hash work?

Our second model is known as a “Perceptual Hash” algorithm[[2]](#footnote-2). The idea is quite simple:

Perceptual Hash is an algorithm that produces a “fingerprint” of an image – a small memory piece. We say that 2 images are similar if they have a similar/identical fingerprint.

A simple implementation (that is very similar to the one we used)

Shrink the image into a very small space (for example 16x16 pixels) and save the new compressed image as the original’s fingerprint.

Comparing fingerprints is quite easy: loop over the fingerprint and compare pixel to pixel.

This gives us results that are good enough in our tests.

# 03 Application

Our application

# 04 Data base

# 05 Technologies

TODO: add list of technologies used

# 06 Final result

The final result is a working

1. Note that our model is a filtering model therefore we want as to optimize False-Positives [↑](#footnote-ref-1)
2. A good example of a working perceptual hash model can be found here <https://www.phash.org/demo/> [↑](#footnote-ref-2)