

# PRÁCTICA IMAGEN

---

## INTRODUCCIÓN

En esta práctica, se ha trabajado en la detección de logos a partir de la utilización de distintos momentos de imagen de cara a analizar y comentar las diferencias entre ellos. Para ello, se ha generado un dataset totalmente personalizado y adaptado a las necesidades de investigación comentadas.

En el conjunto de datos a emplear, se dispone por un lado de imágenes de referencia de cada uno de los logos con los que se va a trabajar, y por otro, un conjunto de datasets con logos en distintas posiciones, orientaciones y con distintos contrastes de color para analizar las características que presenta la utilización de cada uno de los momentos ante cada uno de los escenarios planteados.

Los momentos a emplear, van a ser : Zernike, Hu, SIFT y Orb. Entre ellos, destacar que con Zernike y Hu al estar analizando y estudiando los contornos, lo que se hace de cara a obtener el resultado es lo que se conoce como Shape matching. En cambio, para los casos de SIFT y ORB, tras la extracción de los Keypoints con sus respectivos descriptores, se procede con el feature matching. Ya se verá mejor en la propia implementación la diferencia entre estos dos grupos de momentos.

## ESTUDIO DE MOMENTOS

### 1. Zernike

Dentro de lo que se conoce como feature extraction en el procesamiento de imágenes, el cálculo de momentos de Zernike se emplea sobre todo en la caracterización y cuantificación de las formas de diferentes objetos.

La forma en la que los momentos de Zernike describen las formas de los diferentes componentes de una imagen, puede ser a través de una imagen binaria segmentada (segmented binary image) o a través de los propios límites de los objetos, o lo que también se conoce como contornos. Por lo general, suele ser preferible el utilizar la imagen binaria segmentada, ya que en términos generales es menos susceptible al ruido, pero esto también puede depender mucho del caso de uso.

El proceso de descripción de las formas, a grandes rasgos, se basa en la teoría de las funciones ortogonales, cuyo beneficio reside en la no redundancia de la información entre los diferentes momentos, haciéndolos así más robustos y discriminatorios.

Al final, al hacer uso de los polinomios de Zernike para hacer esta descomposición, se tiene una mejor representación que con el mismo número de coeficientes que una descomposición con polinomios no ortogonales.

Las principales características, en términos generales, serían la robustez al ruido comentada y la invarianza a la rotación.

## **2. Hu**

Zernike es una evolución de Hu, por lo que en términos generales se consideran descriptores de imagen más potentes y más precisos con muy poco coste computacional adicional. A pesar de esto, se ha hecho uso de Hu para ver sus implicaciones y la influencia que puede tener en los resultados.

Básicamente, a modo de resumen, con Hu las imágenes se traducen en 7 momentos bien conocidos, que son invariantes a la rotación, escala y traslación, y que son calculados a partir de los momentos normales. Una vez se obtienen estos, el matching se efectúa a partir del cálculo de estos momentos en cada uno de ellos y haciendo uso de alguna de las funciones de similitud conocidas. Como se verá, la que se ha empleado en este caso ha sido la distancia euclídea.

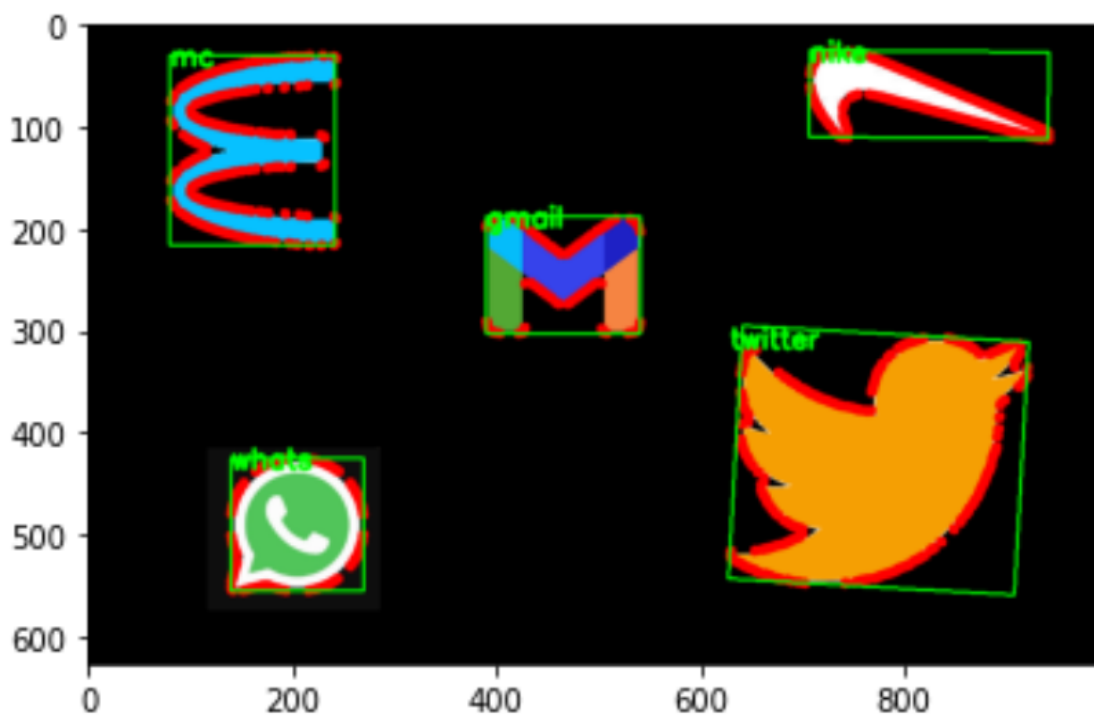
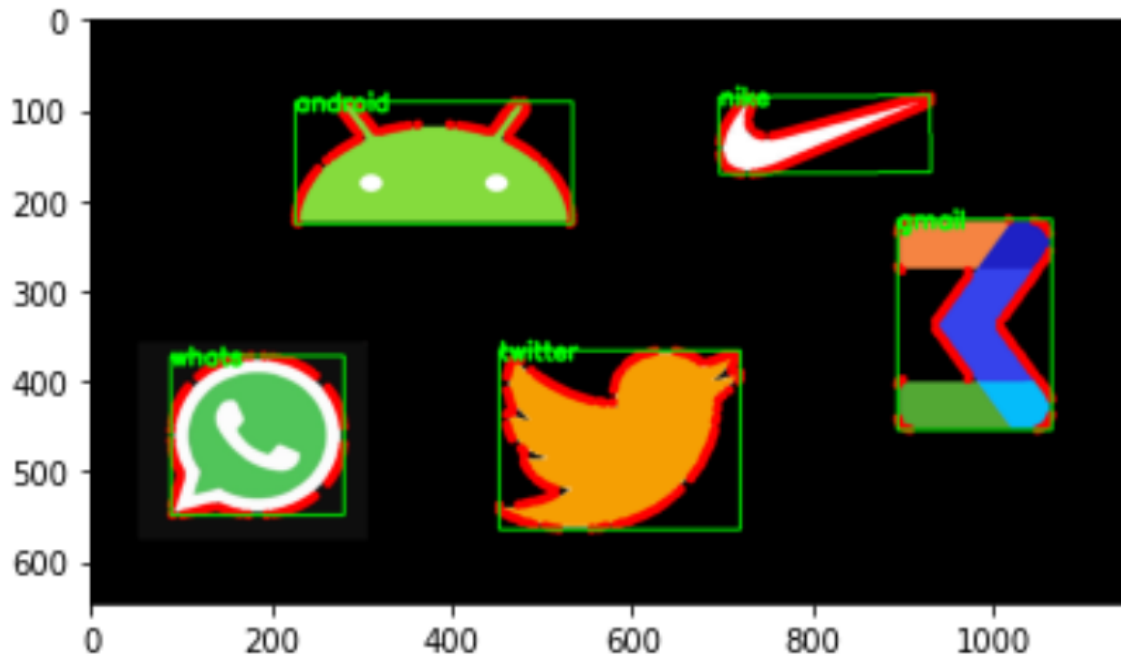
## **METODOLOGÍA ZERNIKE Y HU**

La metodología aplicada para la utilización de estos momentos se basa en la búsqueda de los contornos y el establecimiento de la distancia euclídea entre momentos de la imagen. A grandes rasgos, el procedimiento es el siguiente:

- Preprocesamiento: para tanto imágenes de referencia como las del dataset se realiza un preprocesamiento que abarca una transformación a escala de grises, un blurring y un thresholding.
- Extracción de contornos y momentos de la imagen de los mismos para las imágenes de referencia y las imágenes del dataset.
- Búsqueda de la mínima distancia euclídea: se hace una serie de iteraciones con las que se obtienen las distancias euclídeas entre todos los logos de referencia y todos los contornos de la imagen del dataset
- Determinación de cuál es el logo de referencia más parecido a cada uno de los contornos: a partir de una serie de vectores de mínimas distancias (uno por cada contorno) podemos determinar cuál de los logos de referencia es el más cercano a dicho contorno según que posición de cada vector es el que tiene el valor de distancia euclídea más pequeño.
- Impresión de la imagen con rectángulos que cubren los contornos y el nombre de la compañía cuyo logo es más parecido a dicho contorno.

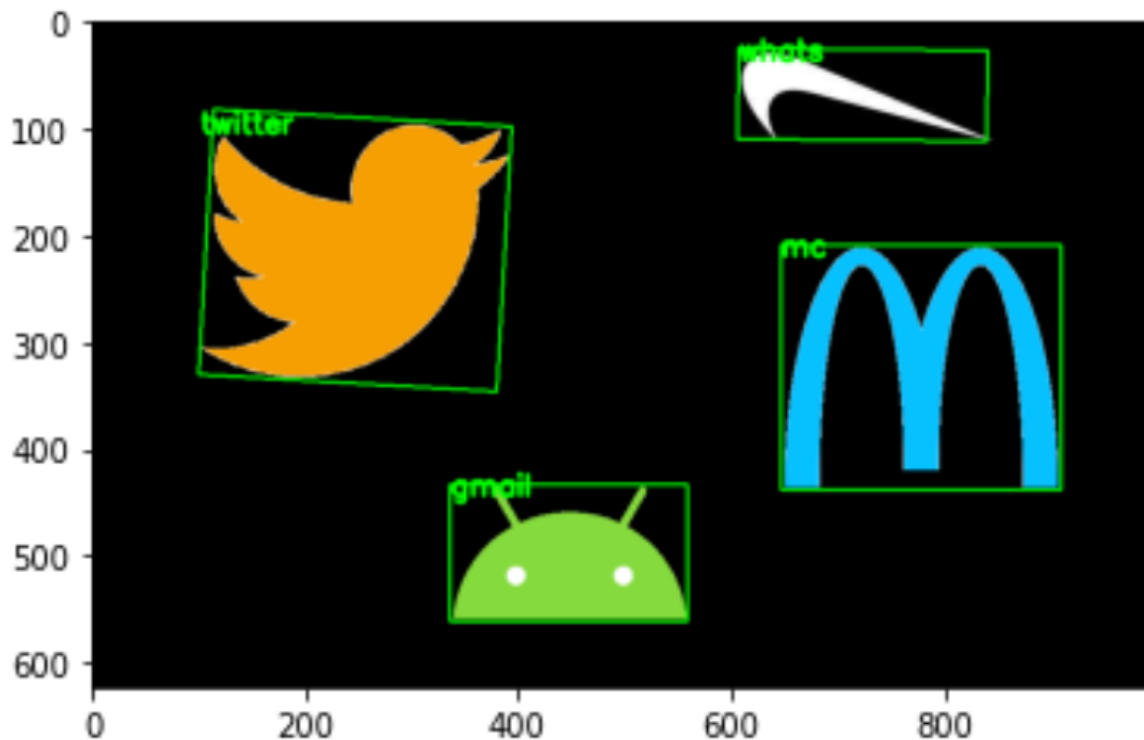
## COMENTARIO DE LOS RESULTADOS OBTENIDOS

A continuación, se muestran algunos de los resultados conseguidos con Zernike en la detección de logos:



En los ejemplos de los resultados de Zernike, se puede apreciar que acierta con todos los logos definidos. Entre una imagen y otra, se pueden apreciar variaciones en la orientación y en cómo esta no influye en el resultado y es capaz de detectar la forma de la imagen y matchearla correctamente con cada una de las imágenes de referencia asociadas.

En cambio, con Hu, se obtienen los siguientes resultados:



En este caso, sí que se aprecian más fallos. Por ejemplo, en la representación que se muestra en la parte superior, se puede ver un claro fallo en la detección de Nike, ya que lo determina como Whatsapp, a pesar de que la forma del logo sea significativamente diferente. En este caso, esta cambiada de orientación, pero en el caso de Android, no es capaz de detectarlo correctamente y asegura que se trata de Gmail, aunque sí que es verdad que es más comprensible la dificultad encontrada en la distinción entre las formas.

La mejora en el rendimiento para Zernike, era de esperar teniendo en cuenta lo que se ha contado en un principio, por lo que se podría decir que los resultados han cumplido con lo esperado en esa detección de logos que se buscaba desde un principio.

### **3. Sift (Scale Invariant Feature Transform)**

El algoritmo SIFT se caracteriza, como su nombre indica, por la invarianza ante las escalas y rotaciones en las imágenes. Aparte, también se considera que dispone de cierta invarianza a la iluminación, la transformación afín y de proyección, lo cual, evidentemente es favorable teniendo en cuenta su finalidad orientada a la descripción de características locales.

Al final, hace uso de un vector de características de una dimensión determinada para describir los Keypoints detectados en la imagen, de forma que esta pueda expresarse directamente como el conjunto de esos vectores asociados a los distintos Keypoints.

Consta de 4 pasos bien definidos previos al propio Keypoint Matching, que se describirán a continuación:

1. Escala de detección de polos espaciales
2. Posicionamiento preciso de los Keypoints
3. Determinación de la dirección de los Keypoints
4. Descripción de los Keypoints como vectores de características.

Uno de los principales defectos del algoritmo SIFT es que se basa demasiado en la dirección del gradiente calculado en los píxeles en esa fase de búsqueda de la dirección principal, lo que puede influir en que esta sea inexacta. Aparte, es considerado un algoritmo matemáticamente complicado y con un alto coste computacional. Al final, está basado en el histograma de los gradientes, calculando los gradientes de cada pixel en el patch, lo que también influye en que no sea muy efectivo en dispositivos que dispongan de una menor capacidad de cómputo.

Al final, a pesar de todo esto, SIFT es considerado el enfoque clásico en términos de feature extraction, además de ser una inspiración para otros algoritmos descriptores, con una mayor precisión y otra serie de características previamente mencionadas.

### **4. ORB**

Oriented FAST and Rotated BRIEF (ORB) fue desarrollado como una alternativa eficiente y viable a SIFT y SURF. ORB se comporta tan bien como SIFT en la tarea de detección de características y es casi dos órdenes de magnitud más rápido. Está basado en el conocido detector de puntos clave **FAST** y en el descriptor **BRIEF**. Ambas técnicas son atractivas por su buen rendimiento y su bajo coste.

- **FAST (Features from Accelerated and Segments Test)**

Dado un píxel  $p$  en una matriz, Fast compara el brillo de  $p$  con los 16 píxeles circundantes que se encuentran en un pequeño círculo alrededor de  $p$ . Los píxeles del círculo se clasifican en tres clases (más claro que  $p$ , más oscuro que  $p$  o similar a  $p$ ). Si más de 8 píxeles son más oscuros o más brillantes que  $p$ , se selecciona como punto clave. Así, los puntos clave encontrados por el método rápido nos dan información sobre la ubicación de los bordes determinantes en una imagen

- **BRIEF (Binary Robust Independent Elementary Feature)**

Brief toma todos los puntos clave encontrados por el algoritmo rápido y los convierte en un vector de características binarias para que juntos puedan representar un objeto. El vector de características binarias, también conocido como descriptor de características binarias, es un vector de características que sólo contiene 1 y 0. En resumen, cada punto clave se describe mediante un vector de características que es una cadena de 128-512 bits.

## **METODOLOGÍA SIFT Y ORB**

Esta se basa en 2 partes:

- Recortes de la imagen: al aplicar SIFT u ORB directamente sobre una imagen del dataset se observa que los keypoints matcheados entre referencia e imagen son muy variados y no consigue elegir los específicos a la referencia usada. Por ello, se ha decidido hacer un paso previo en el que se obtienen las coordenadas de los contornos y se recortan en forma rectangular para luego ser almacenadas en la propia máquina. Sobre cada uno de estos “recortes”, que ya contendrán logos individuales, se aplicarán SIFT u ORB.
- Extracción de features y matching: una vez elegida una imagen de referencia y una de las imágenes guardadas que contienen uno de los logos de la imagen del dataset, se procede a pasar dichas imágenes a escala de grises, se extraen los keypoints y descriptores y se utiliza el BruteForce matching para obtener una imagen en la que se han trazado líneas de referencia a imagen del dataset que unen los keypoints matcheados.

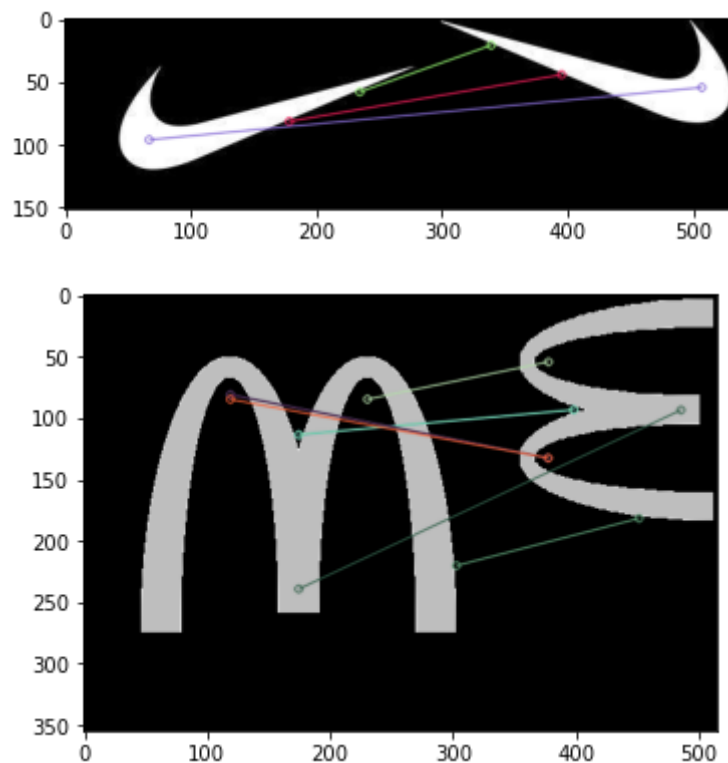
## **COMENTARIO DE LOS RESULTADOS OBTENIDOS**

En el caso de SIFT, estos han sido 2 de los resultados obtenidos para los contornos de Nike y McDonalds:

## Análisis de datos no estructurados : IMAGE

Carlos García-Mauriño Villanueva , Alvaro Ruiz Cornejo y

Carlos García Vázquez



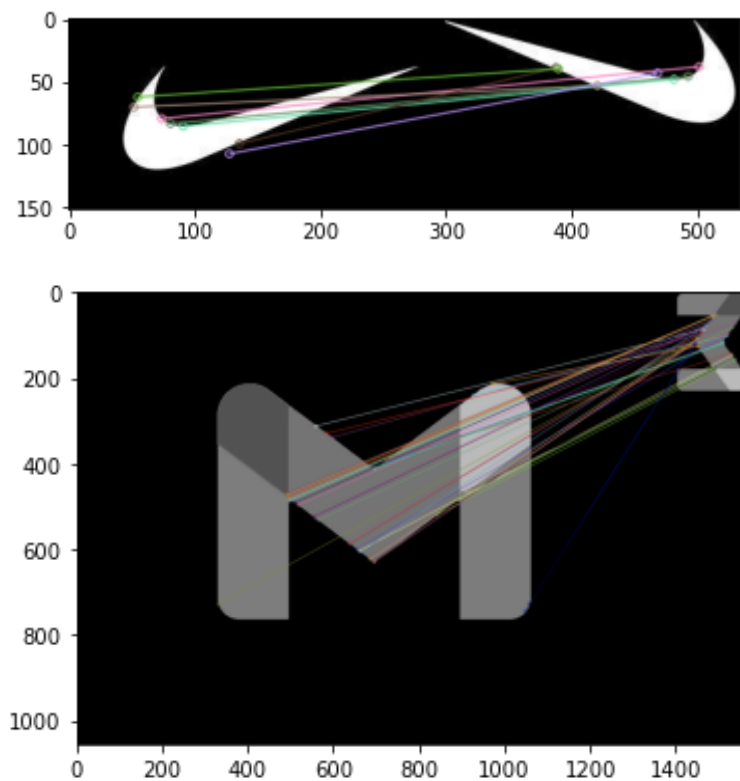
Se puede ver cómo en ambos casos el matching de los Keypoints es prácticamente perfecto, independientemente de la orientación y tamaño de los logos, lo que en cierta forma, era de esperar dadas las características del momento comentadas previamente. Lo único a destacar podría ser el número de Keypoints detectados, que no es demasiado elevado.

Para ORB, en este caso, para Nike también y Gmail, estos han sido los resultados obtenidos:

## Análisis de datos no estructurados : IMAGE

Carlos García-Mauriño Villanueva , Alvaro Ruiz Cornejo y

Carlos García Vázquez



Aquí, sí que es verdad que el número de KeyPoints encontrado es superior al que veíamos en SIFT, pero también se ve que se equivoca más en ese matching realizado entre los features extraídos en la imagen de referencia y la imagen objetivo.

Lo único a destacar como diferencia entre ambos aparte de lo comentado, sería el tiempo de ejecución, que en el caso del ORB ha sido más reducido, como era de esperar desde un principio.