

Trabajo Final

Inteligencia Artificial I

2021

Ferran Martinez
Ingeniería en Mecatrónica

Índice

1. Resumen	1
2. Introducción a la visión artificial	2
3. Agente	4
4. Resultados	5
5. Conclusiones	11
6. Bibliografía	12

1. RESUMEN

Se busca mediante visión artificial la clasificación de los siguientes elementos:

- Tornillos
- Clavos
- Arandelas
- Tuercas

Esto se logra al capturar ciertos valores de interés en las imágenes tomadas, las cuales al pasar por una serie de filtros y cálculos podemos a través de algoritmos de clasificación poder predecir qué tipo de objeto se trata el de la imagen.

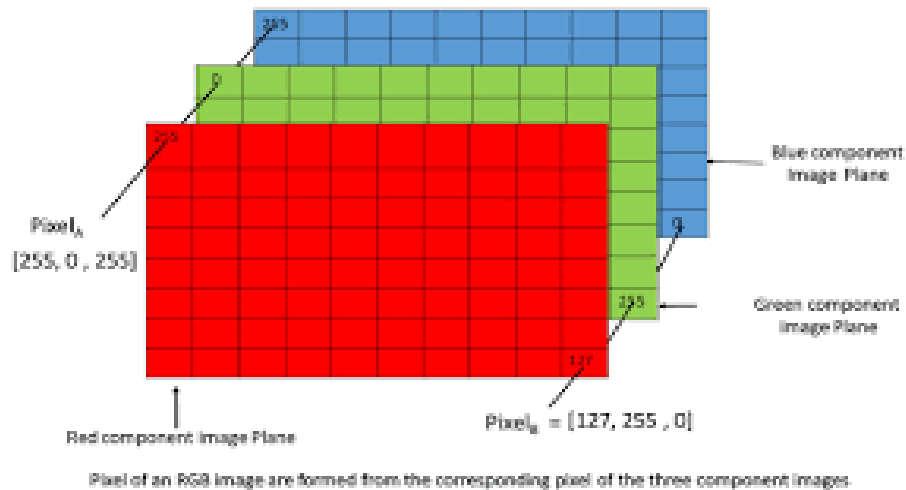
Para ello, se usa una serie de imágenes de elementos que van a formar una base de datos, luego se sustraerán características que nos sirvan para diferenciar unas imágenes de otras y finalmente mediante los algoritmos knn (k nearest neighbor) y k-means lograremos dividir en k grupos (cantidad de ítems distintos) la información de la base de datos.

2. INTRODUCCIÓN

La visión artificial es una disciplina científica que incluye métodos para adquirir, procesar, analizar y comprender las imágenes del mundo real con el fin de producir información numérica o simbólica para que puedan ser tratados por un ordenador.

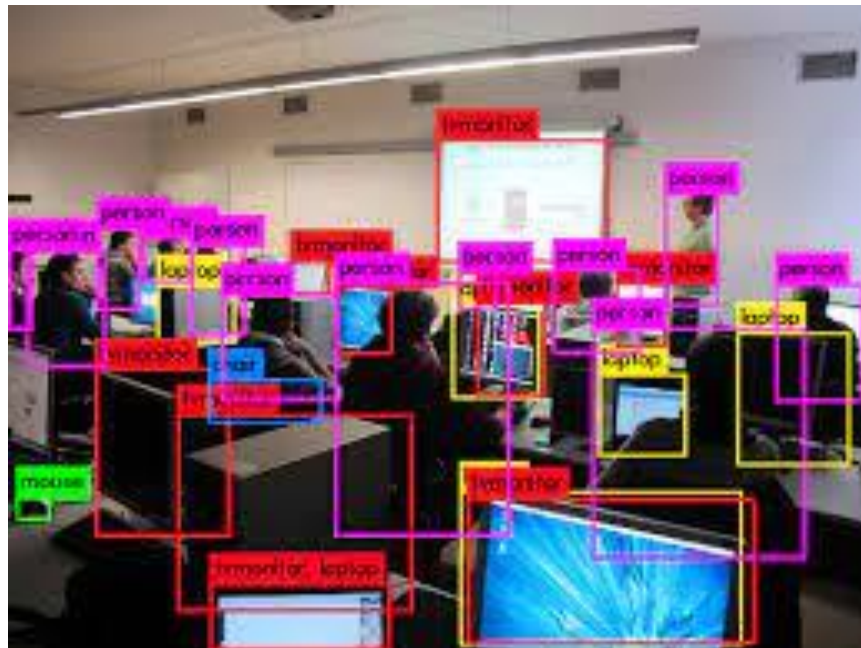
Tal y como los humanos usamos nuestros ojos y cerebros para comprender el mundo que nos rodea, la visión artificial trata de producir el mismo efecto para que los ordenadores puedan percibir y comprender una imagen o secuencia de imágenes y actuar según convenga en una determinada situación.

Las imágenes en una pantalla se visualizan gracias a los píxeles en ella, estos son la menor unidad homogénea en color que forma parte de una imagen digital. La misma se compone de los tres colores primarios (rojo, verde y azul) sobre la misma capa, y con las variaciones de intensidad en cada color podemos componer objetos en el modelo de color RGB.



Entonces para representar una imagen completa a color, se usan tres capas matrices con valores que van del 0 al 255 (máximo valor decimal en 8 bits) donde cada elemento corresponde a un píxel determinado en la pantalla.

Los valores de los elementos de la matriz corresponden a la intensidad del color en el píxel, de esta manera si el mismo elemento en las tres matrices tiene el valor 255, obtendremos el color blanco, y si el valor de cada elemento es 0, el color será negro.



La detección de objetos es la parte de la visión artificial que estudia cómo detectar la presencia de objetos en una imagen. Para ello se distinguen dos partes en el proceso:

- Extracción de características
- Asociar los objetos a dichas características

La extracción de características consiste en aplicar modelos matemáticos a la información aportada por una imagen, buscando de alguna forma que resuman su contenido. A estas características las llamamos descriptores.

Para resolver el problema, debemos enfrentar las siguientes propuestas:

- ¿Cómo se van a representar los objetos?
- ¿De qué manera se van a clasificar? ¿Cuál es la base de datos?
- ¿Cómo se usará el clasificador con objetos nuevos?

3. AGENTE

Trataremos con un agente reactivo basado en modelos, ya que el agente debe manejar un tipo de estado interno que depende de la historia percibida.

Agente	Rendimiento	Actuadores	Sensores
Modelo de visión artificial	Predicción correcta por parte de los clasificadores	Filtros lineales, cálculos de descriptores y clasificadores (algoritmos)	Cámara del teléfono

Propiedades del entorno de trabajo:

- Fondo blanco, minimizando lo máximo posible las sombras sobre el mismo.
- Totalmente observable
- Episódico
- Estático
- Discreto

4. RESULTADOS

Como se dijo anteriormente, para iniciar con el proceso de visión artificial, debemos resolver el problema de la representación, es decir, que representan las imágenes.

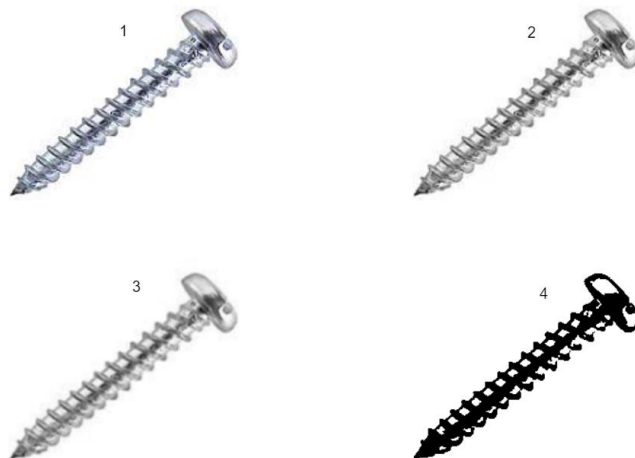
Tomando como referencia una imagen de la base de datos:



Podemos distinguir fácilmente que se trata de un tornillo, pero, ¿cómo hacemos para que el agente reconozca dicho objeto, como podemos representar la imagen en lenguaje de máquina?

Para ello aplicamos el cálculo de momentos en imágenes, pero antes de ello, tenemos que segmentar la imagen para pasar de 3 matrices con valores que van desde 0 a 255 a una sola matriz con solo dos valores, 1 y 0.

Esto se logra aplicando una serie de filtros que ayudarán a remarcar la forma del objeto.



El proceso de filtros consiste:

1. Cambio a escala de grises (se promedian las intensidades de las 3 capas pixel a pixel)
2. Se aplica un desenfoque gaussiano a través de una matriz de convolución 3x3
3. Se binariza la imagen, es decir que bajo cierto umbral se determina si corresponde un máximo valor o uno mínimo (cero y uno análogos a 255 y 0)
4. Finalmente se aplica otra matriz de convolución para realzar los bordes de la imagen, esto ayuda a eliminar ciertos ruidos en piezas como los tornillos.

Una vez finalizado el proceso de aplicación de filtros, podemos calcular los momentos de la imagen.

Por definición, un momento de una imagen es cierto promedio ponderado particular de las intensidades de los píxeles de una imagen. Los momentos espaciales de orden $p + q$ de una imagen $I(x,y)$ son:

$$M_{p,q} = \sum_{x=1}^N \sum_{y=1}^M x^p y^q I(x, y).$$

En nuestro caso, el orden del momento será 4, ya que así lograremos una matriz 3x3 que nos servirá para los próximos pasos.

Posteriormente se calculan los momentos centrales de la imagen. Estos los podemos sacar directamente a partir de las siguientes formulas derivadas de la matriz de momento espacial:

$$\mu_{0,0} = M_{0,0}$$

$$\mu_{0,1} = 0$$

$$\mu_{1,0} = 0$$

$$\mu_{2,0} = M_{2,0} - \frac{M_{1,0}^2}{M_{0,0}}$$

$$\mu_{1,1} = M_{1,1} - \frac{M_{1,0}M_{0,1}}{M_{0,0}}$$

$$\mu_{0,2} = M_{0,2} - \frac{M_{0,1}^2}{M_{0,0}}.$$

Luego, los mismos se normalizan para obtener una descripción independiente del tamaño:

$$\eta_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}^\alpha} \quad \alpha = \frac{p+q}{2} + 1.$$

A partir de los momentos centrales normalizados de orden dos y tres es posible extraer siete parámetros denominados momentos invariantes de Hu, independientes a posición, tamaño y ángulo del objeto:

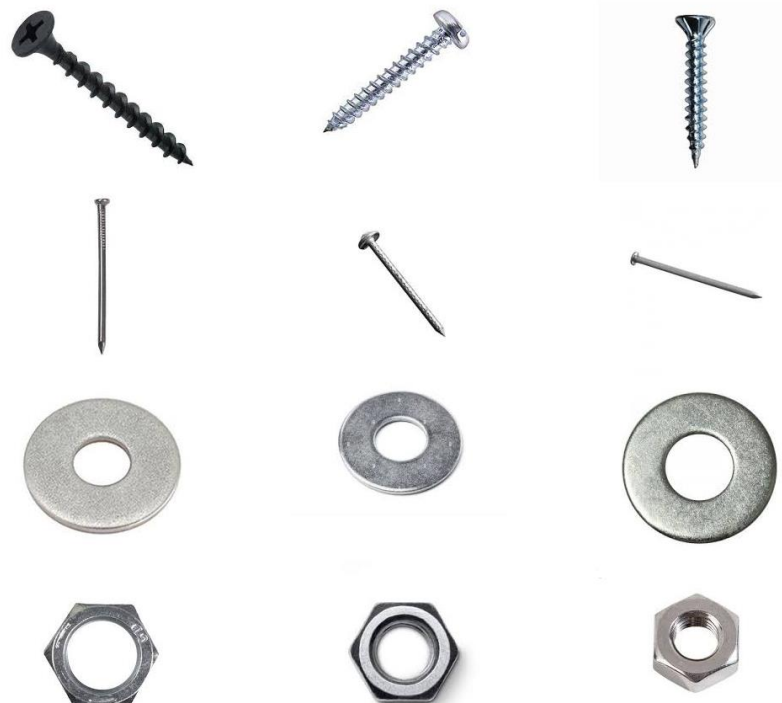
$$h_1 = \eta_{2,0} + \eta_{0,2}$$

$$h_2 = (\eta_{2,0} - \eta_{0,2})^2 + 4\eta_{1,1}^2$$

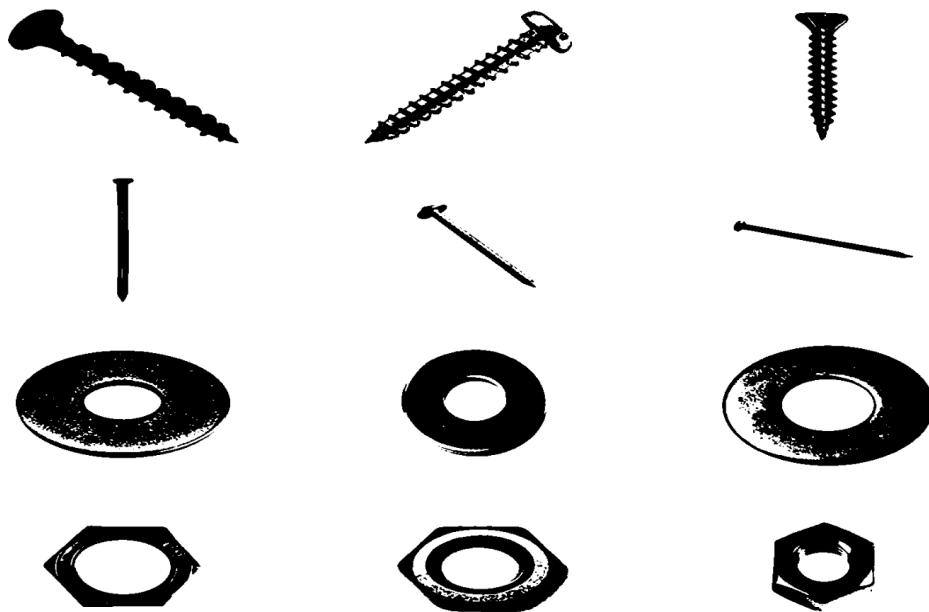
Finalmente, con estos dos datos procedemos a generar una gráfica donde

- $1/h_1$ = valor en x
- $1/h_2$ = valor en y

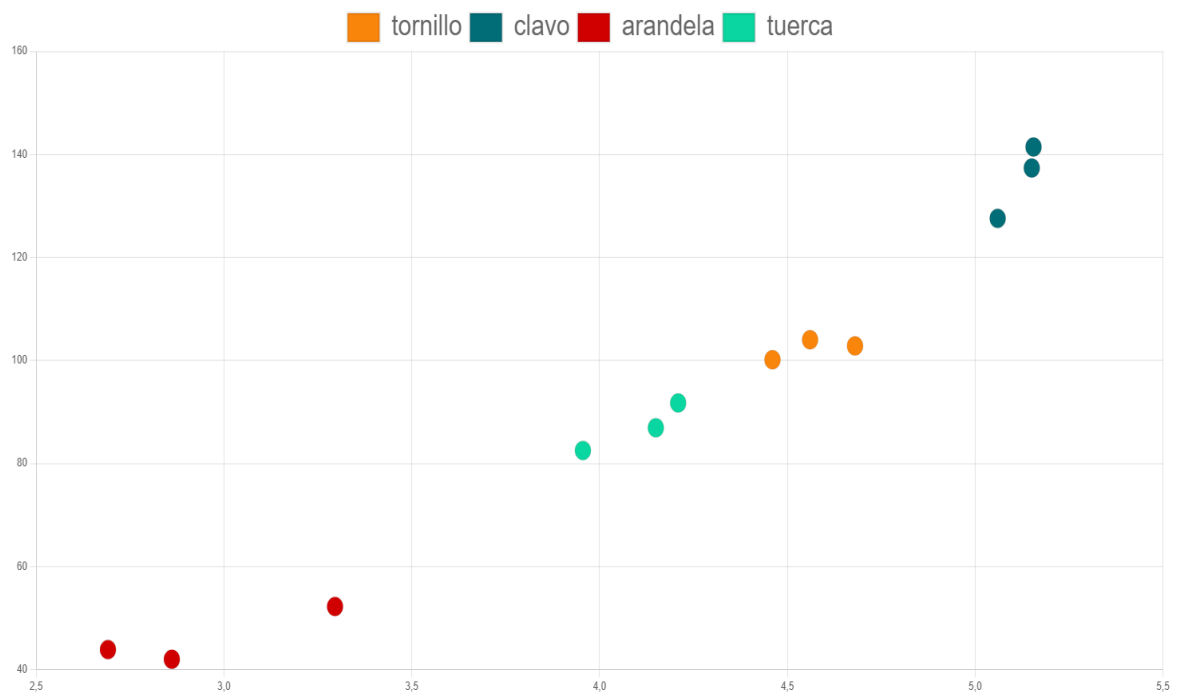
Aplicando el proceso a nuestra base de datos conocida



Se obtienen las imágenes con los filtros correspondientes listas para sustraer las características



Y finalmente se plasman los datos en una gráfica para reconocer los distintos grupos



En la gráfica podemos observar lo relevante que ha sido el proceso, ya que conseguimos separar en 4 distinguidos grupos cada uno de los diferentes objetos analizados.

Por lo tanto, hemos logrado representar los objetos para que el agente pueda interpretarlos de forma correcta y además genero nuestra base de datos ya clasificada, donde el agente ya reconoce cuales datos corresponden a que grupo.

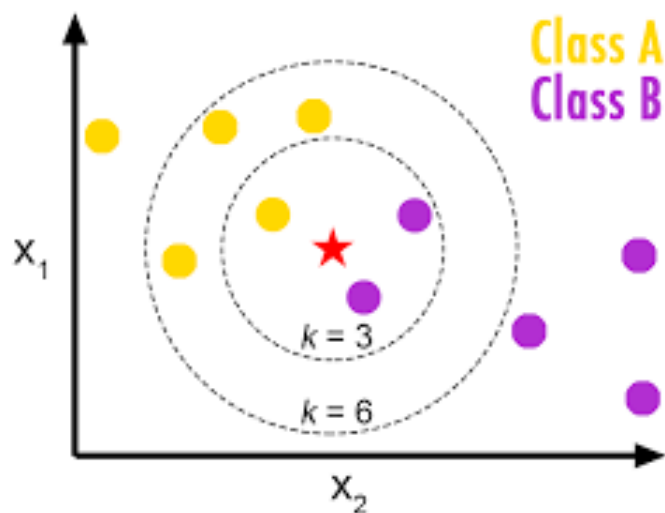
Ahora solo nos queda buscar una herramienta que pueda clasificar un nuevo objeto (previamente segmentado y sustraídas sus características).

Para ello haremos uso de dos algoritmos de clasificación:

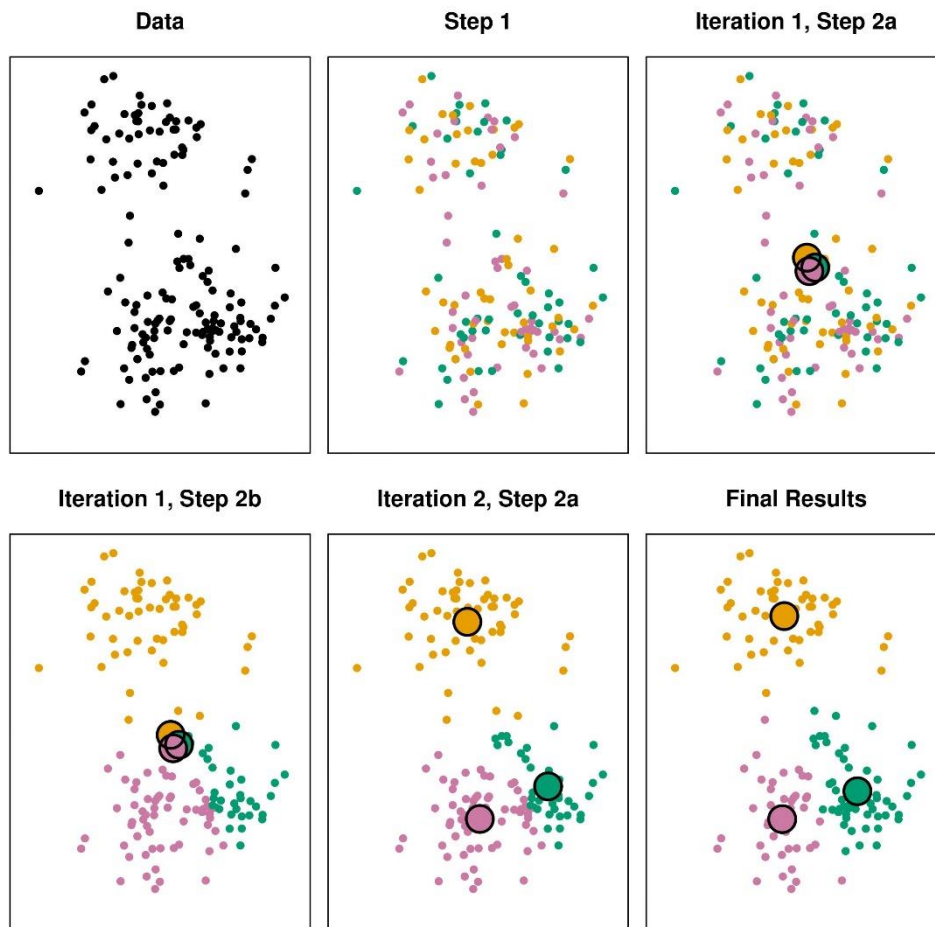
- KNN (k-nearest neighbor)
- K-MEANS (k-medias)

Ambos trabajan sobre una base de puntos ya establecida (de la gráfica anterior) con los grupos definidos.

Cuando se introduce un nuevo objeto (punto en la gráfica), knn buscará una cantidad k de los objetos más cercanos a dicho nuevo objeto, de dicha cantidad evaluará cual es tipo de objeto más predominante y en base a ello el algoritmo determinará a que grupo pertenece el nuevo objeto.



El segundo algoritmo trabaja en función de una posición media dada por cada uno de los grupos ya preestablecidos, dicha media tiene coordenadas en la gráfica y se llama centroide.



Cuando se agregan nuevos objetos, se busca cual es el centroide más cercano a dicho objeto y una vez encontrado, se asigna dicho centroide al objeto (y por ende se asigna el grupo). Posteriormente la posición del centroide se actualiza ya que al agregarse un nuevo dato, se debe volver a calcular la media que da la posición del centroide.

5. CONCLUSIONES

Se concluye que los algoritmos de clasificación son precisos y ayudan mucho con estos tipos de tarea, en los que a partir de una base de datos y diferenciando distintos grupos (según sus características) podemos llegar a incluir nuevos datos y predecir a que grupo corresponde.

Sin embargo, se debe tener especial cuidado con los datos que usa inicialmente el agente, ya que, si no se encuentran bien diferenciados, los clasificadores pueden llegar a equivocarse a la hora de adjudicarle un grupo a un nuevo objeto. Para ello recomiendo:

- Que los datos de entrenamiento correspondan lo máximo posible con los datos que vayan a introducirse al modelo. En mi caso usé fotos de stock como base de datos, y a la hora de realizar pruebas ingresando imágenes capturadas de forma sencilla, los datos estaban bastante desviados, causando alguna mala predicción por parte del algoritmo.
- Usar una elevada cantidad de información para entrenar al algoritmo (base de datos). Como vimos, cuanto mayor es la base de datos, mayor es la cantidad de puntos en nuestra gráfica y por lo tanto más referencias tienen los algoritmos a la hora de ponderar y sacar conclusiones.

En el caso de este trabajo se utilizó el lenguaje web JavaScript para realizar la parte lógica del programa, por otro lado, para la visualización de las imágenes y gráficas, se usaron los lenguajes HTML y CSS, también lenguajes web.

A continuación, se deja el código fuente del trabajo y el despliegue web para poder visitar el sitio (y probar los algoritmos):

Código fuente: https://github.com/Feeeeerran/IA-Trabajo_final

Despliegue web: https://feeeeerran.github.io/IA-Trabajo_final

6. BIBLIOGRAFÍA

<https://www.cs.princeton.edu/courses/archive/spring07/cos424/lectures/liguest-lecture.pdf>

<http://poseidon.tel.uva.es/~carlos/ltif10001/descriptores.pdf>

https://developer.mozilla.org/es/docs/Web/API/Canvas_API/Tutorial/Pixel_manipulation_with_canvas

<https://themachinelearners.com/algoritmo-knn/>

https://www.unioviedo.es/compnum/laboratorios_py/kmeans/kmeans.html

https://es.wikipedia.org/wiki/Procesamiento_digital_de_imágenes