

# Trabajo Final

## Sistema clasificador de piezas mediante visión artificial

Universidad Nacional de Cuyo  
Inteligencia Artificial I  
2020

Costarelli Ignacio Agustin  
Legajo 10966  
agustin@costarellisa.com.ar

## Contenido

<b>Resumen .....</b>	<b>2</b>
<b>Introducción.....</b>	<b>3</b>
<b>Diseño del Sistema.....</b>	<b>4</b>
<b>Ejemplo de aplicación .....</b>	<b>9</b>
<b>Conclusiones. ....</b>	<b>13</b>
<b>Bibliografía .....</b>	<b>14</b>

## Resumen

El siguiente trabajo consiste en un sistema clasificador de piezas, mediante visión artificial se podrá diferenciar entre arandelas, clavos, tuercas y tornillos. El mismo se desarrolló utilizando los algoritmos k-means (no supervisado) y knn (supervisado).

Para el tratamiento de las imágenes fue utilizado MATLAB donde también se desarrolló el código para el funcionamiento del programa.

Por último se realizan pruebas del código realizado para comprobar su eficiencia

## Introducción

El propósito del siguiente trabajo es diseñar un sistema el cual mediante visión artificial se pueda reconocer y distinguir entre arandelas, clavos, tornillos y tuercas. Este mismo sistema podría ser llevado a cualquier industria que trabaje con estos elementos y requiera un reconocimiento automático de los mismos.

El agente desarrollado se planteó como un agente que aprende debido a que a medida que desarrolla la tarea de identificación va aumentando la base de dato la cual es la base de conocimiento del sistema. A medida que desarrolle la tarea ira mejorando la misma debido al incremento de dicha base.

Los algoritmos diseñados para este agente son dos, el k-means el cual es un algoritmo no supervisado y el knn que es un algoritmo supervisado.

La diferencia entre ambos es que el algoritmo supervisado posee una base de datos a la cual se la clasifica cuando se introduce y el algoritmo no supervisado recibe una base de datos la cual no tiene ninguna clasificación previa.

Tablas REAS.

<b>Tipo de agente</b>	<b>Medida de rendimiento</b>	<b>Entorno</b>	<b>Actuadores</b>	<b>Sensores</b>
Sistema de reconocimiento de arandelas, clavos, tornillos y tuercas.	Mayor cantidad de piezas reconocidas exitosamente	Fabrica, operarios.	Cinta transportadora, iluminación automática.	Cámara de visión artificial

Las propiedades del entorno de trabajo son las siguientes

- Totalmente observable
- Estocástico
- Episódico
- Estático
- Discreto
- Agente individual

## Diseño del Sistema

### Obtención de imágenes.

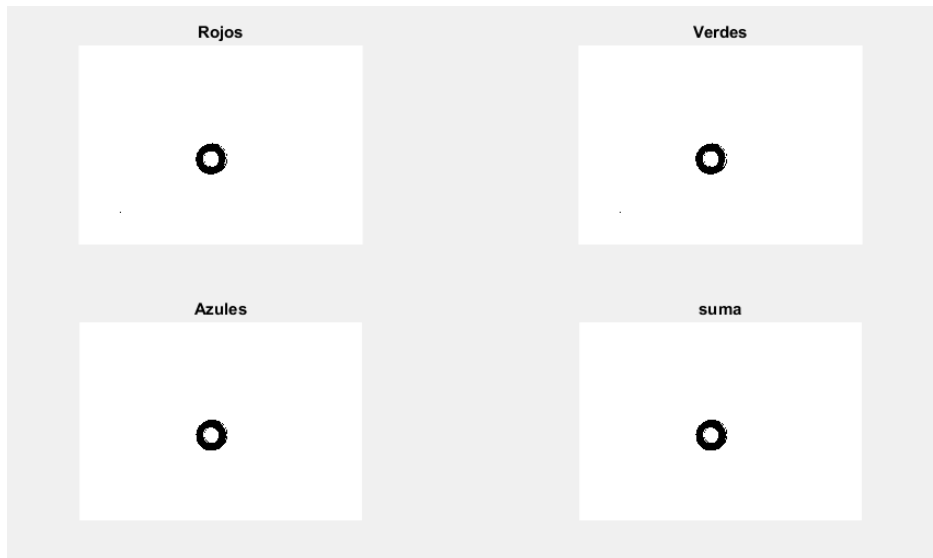
La fotografía de las piezas se trató de realizar siempre con las mismas condiciones para evitar la introducción de perturbaciones externas al sistema. Todas las fotos fueron realizadas en una caja con iluminación y fondo blanco. La mayor dificultad de esta parte fue que las fotos siempre tuvieran una iluminación uniforme para evitar futuros errores en el tratamiento de las imágenes.

### Tratamiento de imágenes.

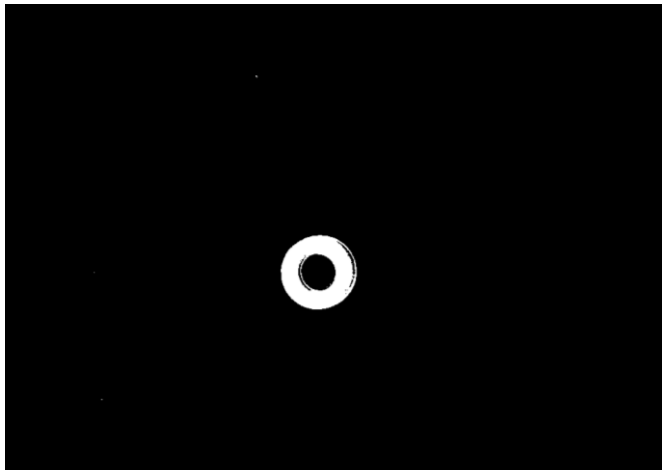
1- Se partió de la imagen tomada con la cámara.



2- La fotografía posee 3 capas una correspondiente a cada uno de los colores primarios de la luz (azul, rojo, verde), primero se separaba cada una de esas capas y se las transformaba a su forma binaria, la cual es una matriz formada por 0 y 1 que al verse reflejada como imagen son pixeles blancos y negros. Luego de obtener las 3 capas en su forma binaria se procedía a hacer la suma de las mismas para obtener una imagen binaria que representara mejor al objeto.

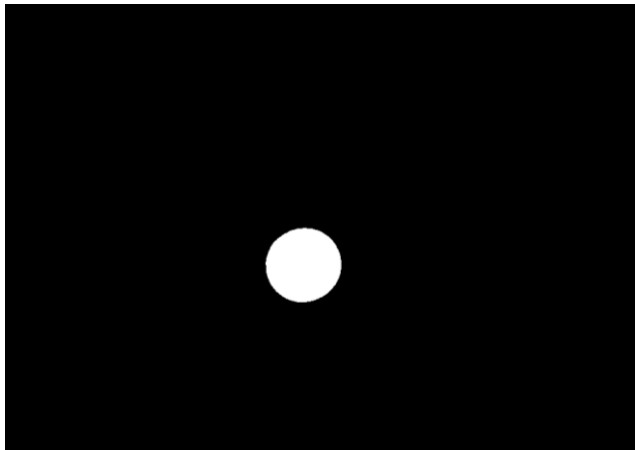


3- Se invierten los colores blanco y negro de la imagen para tener un mejor elemento de trabajo.



4- Todos los agujeros de la imagen son rellenados para tapar posibles pequeños ruidos y eliminar el agujero de la arandela y tuerca.

5- Por ultimo para formar una pieza con forma más suave mediante una función que rellena con pequeños discos la figura con lo cual se pierden bordes agudos y se corrigen algunos contornos irregulares.



Cuando la iluminación no era uniforme el resultado era el siguiente.



Si por error se analiza esta imagen los parámetros extraídos harán que los resultados del algoritmo sean muy distintos a lo esperado.

### Obtención de parámetros.

Con la imagen ya procesada se procedió a buscar parámetros característicos que nos pudieran brindar la información para distinguir entre cada objeto. Se utilizó la función regionprops de Matlab. Se hicieron pruebas buscando relación con área, perímetro, excentricidad, máxima longitud axial, mínima longitud axial, redondez, diámetro equivalente de un círculo con la misma área que la pieza.

Observando los valores de distintas imágenes se concluyó que dos parámetros que podría ser utilizados para resolver el problema eran la excentricidad y la división entre la longitud axial max y min.

Si bien esos dos parámetros ayudaban a diferenciar bien las piezas, seguía habiendo valores muy similares entre clavos y tornillos para evitar posibles errores también se adoptaron los momentos de Hu.

Los momentos de hu son 7 descriptores en función de los momentos geométricos de una imagen. Los cuales son invariantes a la rotación y escalamiento. Para nuestra aplicación solo se calcularon 2 de los momentos de Hu.

objeto	Excentricidad	división	momento 1	momento 2
arandela	0,251203711	1,03312803	0,15924482	2,69E-05

arandela	0,290219573	1,04497567	0,15944626	4,91E-05
clavo	0,997842888	15,2329155	1,36335639	1,82702142
clavo	0,999043495	22,8689041	1,99310169	3,9422239
tornillo	0,979585398	4,9744197	0,50001712	0,2126867
tornillo	0,961882213	3,65678725	0,37234057	0,10274418
tuercas	0,334675741	1,06119559	0,16029104	9,05E-05
tuercas	0,310308465	1,05192743	0,16029633	6,58E-05

### Algoritmo K-means.

Con toda la información extraída de las piezas se implementó el algoritmo K-means. Es no supervisado dado que los datos no tienen etiquetas y estos se clasifican a partir de su estructura interna (propiedades, características).

Consiste en agrupar objetos en  $k$  grupos basándose en sus características. El agrupamiento se realiza minimizando la suma de distancias euclidianas entre cada objeto y el centroide de su cluster.

El algoritmo consta de tres pasos:

1. **Inicialización:** una vez escogido el número de  $k$  clusters, se establecen  $k$  centroides en el espacio de los datos. Para esta aplicación el primer centroide se ubicó utilizando valores similares a los obtenidos en la extracción de características de los objetos.
2. **Asignación objetos a los centroides:** cada objeto de los datos es asignado a su centroide más cercano mediante el cálculo de la distancia euclidiana
3. **Actualización centroides:** se actualiza la posición del centroide de cada grupo tomando como nuevo centroide la posición del promedio de los objetos pertenecientes a dicho grupo.

La implementación del algoritmo consistió en introducir una base de 30 fotos, se les realizó el procesamiento y extracción de características. Para este caso se definieron 4 clusters, luego se midió en cada objeto la distancia a los centroides establecidos. A medida que se encontraba la menor distancia era introducida la información del objeto junto a una etiqueta en una de cuatro matrices. A cada matriz se le calculó la media para cada una de las características y determinar un nuevo punto para cada centroide. Este proceso se repitió 3 veces más para obtener centroides más precisos que sirviera para la clasificación.

Con esta base de datos cada vez que se quiera analizar un objeto nuevo, recibirá el mismo procesamiento de imagen, extracción de características y se calculará su distancia a los centroides para determinar que etiqueta le corresponde.

### Algoritmo K-nn.

La otra implementación de la información recaudada fue mediante el algoritmo K-nn el cual se trata de un algoritmo de aprendizaje supervisado debido a que disponemos de un conjunto de datos que llamamos datos de entrenamiento los cuales serán la información de partida para la futura implementación del algoritmo.



El funcionamiento del algoritmo es:

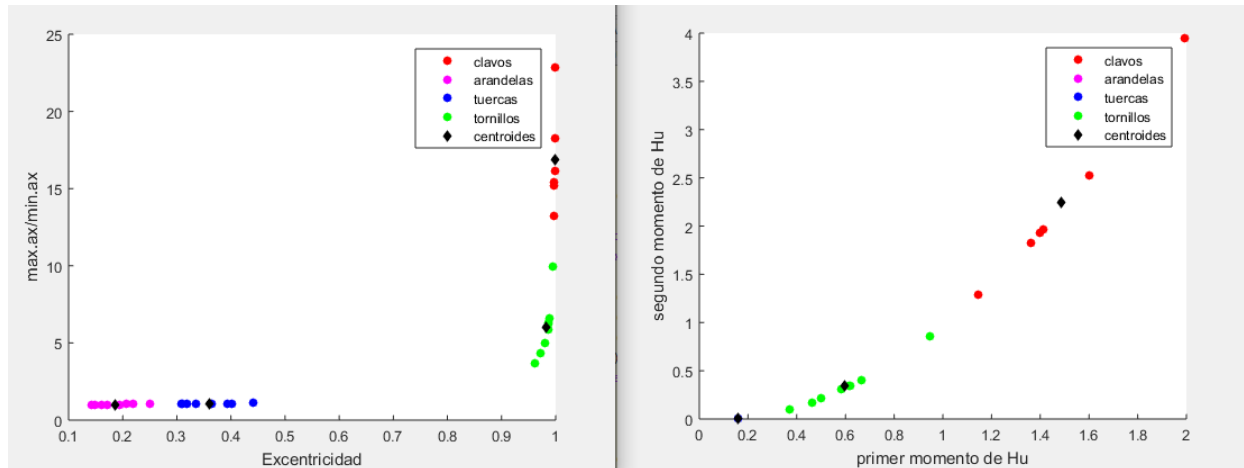
1. Tomar un punto que representado por las características de una imagen dada y calcular la distancia a todos los demás puntos de la base de datos.
2. Seleccionar los  $K$  puntos más cercanos y observar su clasificación

El valor de  $K$  se elige impar para evitar empate entre la cantidad de vecinos cercanos. Su valor dependerá de la problemática y se determina mediante prueba y error.

Esta implementación consistió en usar el mismo procedimiento de procesamiento de imágenes y la extracción de características ya mencionadas. Primero se dividió la base de fotos en base de datos inicial y datos de entrenamiento. A cada conjunto de características de la base de datos inicial se le colocó la etiqueta correspondiente a cada objeto, luego con los datos de entrenamiento se clasificaron mediante el uso de knn. Se revisaron los resultados, en caso de ser satisfactorios los datos de entrenamiento pasan a ser parte de la base de datos. Por último se prueba el algoritmo utilizando piezas nuevas para comprobar la eficiencia de su funcionamiento.

## Ejemplo de aplicación

### Prueba de K-means



En estas grafías se puede observar la información obtenida de procesar 30 objetos. Debido a que para la aplicación se están usando 4 parámetros la interpretación de un gráfico 4-d sería compleja. Por eso se dividió en dos gráficos que pueden ayudar a brindar una mejor interpretación. En el primero podemos observar una buena separación entre los centroides, pero cuando fue llevado a la práctica con solo dos parámetros el hecho de que los centroides de clavos y tornillos estén prácticamente en la misma línea vertical generó errores de clasificación. Por eso se adoptó los parámetros de la segunda imagen que si bien generan ruido entre los datos de arandelas y tuercas al tener centroides tan cerca la distancia entre clavos y tornillos fue arreglada. La combinación de todos los parámetros brindó mejores resultados.

Con la información de donde se encuentran los centroides se procedió a la clasificación de las siguientes 4 imágenes.



1.jpeg



2.jpeg



3.jpeg



4.jpeg

```
La imagen
1.jpeg
es una tuerca
La imagen
2.jpeg
es un tornillo
La imagen
3.jpeg
es una arandela
La imagen
4.jpeg
es un clavo
fx >>
```

Para las imágenes evaluadas el programa ha demostrado resultados positivos logrando el objetivo deseado.

### Prueba algoritmo Knn

Como primer paso se dividió el conjunto de fotos en 21 para base de datos y 9 para entrenamiento. Las recomendaciones encontradas es que la relación tiene que ser 70/30. Luego de procesarlas y extraer las características se obtuvo la siguiente matriz con etiquetas.

0.1621	1.0134	0.1592	0.0000	1.0000
0.2070	1.0221	0.1592	0.0000	1.0000
0.1419	1.0102	0.1592	0.0000	1.0000
0.2184	1.0247	0.1592	0.0000	1.0000
0.2512	1.0331	0.1592	0.0000	1.0000
0.1482	1.0112	0.1592	0.0000	1.0000
0.9950	9.9779	0.9456	0.8589	4.0000
0.9978	15.2329	1.3634	1.8270	4.0000
0.9990	22.8689	1.9931	3.9422	4.0000
0.9985	18.2829	1.6002	2.5302	4.0000
0.9979	15.4193	1.4121	1.9610	4.0000
0.9855	5.8989	0.6189	0.3414	3.0000
0.9885	6.6144	0.6678	0.4070	3.0000
0.9796	4.9744	0.5000	0.2127	3.0000
0.9619	3.6568	0.3723	0.1027	3.0000
0.9873	6.2908	0.6161	0.3431	3.0000
0.3646	1.0739	0.1604	0.0001	2.0000
0.3347	1.0612	0.1603	0.0001	2.0000
0.3103	1.0519	0.1603	0.0001	2.0000
0.3944	1.0882	0.1605	0.0002	2.0000
0.3190	1.0551	0.1603	0.0001	2.0000

La primer columna representa la excentricidad, la segunda la división de max y min axial, la tercera y cuarta son los momentos de Hu y la cuarta corresponde a la etiqueta donde 1 es arandela, 2 es tuerca, 3 es tornillo y 4 es clavo.

El siguiente paso fue aplicar el algoritmo de Knn a las fotos de entrenamiento. El k utilizado fue 3, dando como resultado lo siguiente.

```
La imagen analizada es:  
A6.jpeg  
El objeto es una arandela  
La imagen analizada es:  
A7.jpeg  
El objeto es una arandela  
La imagen analizada es:  
C5.jpeg  
El objeto es un clavo  
La imagen analizada es:  
C6.jpeg  
El objeto es un clavo  
La imagen analizada es:  
To7.jpeg  
El objeto es un tornillo  
La imagen analizada es:  
To8.jpeg  
El objeto es un tornillo  
La imagen analizada es:  
Tu6.jpeg  
El objeto es una tuerca  
La imagen analizada es:  
Tu7.jpeg  
El objeto es una tuerca  
La imagen analizada es:  
Tu8.jpeg  
El objeto es una tuerca  
v >>
```

Habiendo tenido resultados favorables se procedió a incorporar las fotos de entrenamiento a la base de datos definitiva del algoritmo knn.

Por último se evaluaron nuevamente las 4 fotos que ya habían sido evaluadas por km.



1.jpeg



2.jpeg



3.jpeg



4.jpeg

```
La imagen analizada es:  
1.jpeg  
El objeto es una tuerca  
La imagen analizada es:  
2.jpeg  
El objeto es un tornillo  
La imagen analizada es:  
3.jpeg  
El objeto es una arandela  
La imagen analizada es:  
4.jpeg  
El objeto es un clavo  
>>
```

Como resultado se obtuvo una clasificación exitosa de las 4 fotos.

## Conclusiones.

En la realización del trabajo integrador se pudo observar diferencias entre los algoritmos de aprendizaje supervisados y no supervisados. En ambos se pudo ver la capacidad que poseen para clasificar objetos en función de parámetros.

La primera dificultad encontrada fue por el método de procesamiento de imágenes elegido el cual requería siempre una iluminación uniforme para evitar grandes ruidos en la imagen. Esto llevado a condiciones reales puede resultar un problema por lo tanto es un punto a mejorar.

A la hora de elegir los parámetros, como solo 2 no eran suficientes para poder diferenciar tornillos de clavos, se tuvo que investigar más en profundidad las opciones por fuera de las funciones que brinda Matlab. Como resultado se tomó la decisión de usar los momentos invariantes de Hu los cuales tienen mucho uso en el campo de la visión artificial para clasificar objetos.

Por ultimo hay que mencionar lo útiles que pueden resultar estos algoritmos para trabajos de reconocimiento no muy complejos. Son algoritmos sencillos que requieren poco procesamiento computacional. La problemática que tienen es en caso de tener objetos con características similares y valores muy dispersos, los clusters pierden precisión y a la hora de reconocer objetos los resultados pueden ser menos confiables.

## Bibliografía

-Stuart Russell, Peter Norvig <<Inteligencia Artificial, Un Enfoque Moderno>> PEARSON EDUCACIÓN, S.A., Madrid, 2004

-Josh Starmer (canal de youtube: StatQuest with Josh Starmer)  
StatQuest: K-means clustering.

-Josh Starmer (canal de youtube: StatQuest with Josh Starmer)  
StatQuest: K-nearest neighbors, Clearly Explained.

-Gomez Flores, Wilfrido (2015): Reconocimiento de objetos en fotografías.