

## Clasificador automático de elementos mecánicos empleando Redes Neuronales

### Automatic element classifier of mechanical elements using Neural Networks

PERÉZ-CHIMAL, Rosa Janette†\*, LUNA-PUENTE, Rafael, GUERRERO-ORDAZ, Salvador y JAMAICA-GONZÁLEZ, Alejandro

Universidad Tecnológica de Salamanca, Av. Universidad Tecnológica #200, Ciudad Bajío, Salamanca, Gto. C.P. 36766  
Universidad Tecnológica de Querétaro, Av. Pie de la Cuesta #2501, Unidad Nacional, Querétaro, Qro., México C.P. 76148

ID 1<sup>er</sup> Autor: Rosa Janette, Pérez-Chimal

ID 1<sup>er</sup> Coautor: Rafael, Luna-Puente

ID 2<sup>do</sup> Coautor: Salvador, Guerrero-Ordaz

ID 3<sup>er</sup> Coautor: Alejandro, Jamaica-González

Recibido: Junio 08, 2018; Aceptado Agosto 09, 2018

#### Resumen

Los modelos de redes neuronales - cuyo funcionamiento se basa en el comportamiento de las neuronas del cerebro humano - se han aplicado de manera exitosa en la solución de diversos problemas enfocados al reconocimiento de patrones y la manipulación de abundantes variables. El presente proyecto muestra la clasificación de elementos mecánicos utilizados en la industria para control de calidad de distintos tipos de formas como tornillos, los cuales pueden ser clasificados por tamaño y forma y a su vez poder detectar anomalías de fabricación separando tornillos defectuosos dentro de una producción. Este procedimiento descrito se logra en base al entrenamiento de redes neuronales. Existen actualmente muchos procesos de carácter industrial que requieren de una labor de clasificación, actividad que demanda tiempo, convirtiéndose en un proceso costoso y en ocasiones ineficiente. Se presentan requerimientos para el procesamiento de datos y la implementación de distintos elementos como sensores y motores eléctricos. Así como elementos mecánicos y de manufactura para la elaboración de dispositivos automatizados y sistemas de visión. Se presenta un dispositivo para clasificar una serie de elementos mecánicos por medio de procesamiento de imágenes utilizando inteligencia artificial y la programación de un microcontrolador, el cual consiste en una banda transportadora, una cámara digital y un contenedor giratorio. Usando estos elementos se realiza la clasificación de imágenes, para este caso de tornillos. La plataforma giratoria está sincronizada con una cámara digital para su posterior movimiento angular. Los ángulos son constantes de acuerdo con la clase de los elementos a almacenar, dicha clasificación de tornillos presenta errores promedio del 0.3%.

#### Automatización, Clasificadora, Redes neuronales

#### Abstract

Models of Neural Networks (NN)-whose operation is based on the behavior of neurons of human brain- have been successfully applied to solve several problems focused on pattern recognition and the manipulation of numerous variables. This work shows on the classification of mechanical elements (like screws) used in the industry for its quality control, such components can be classified by size and shape, even it is possible to detect manufacture abnormalities separating the damage screws. The procedure described is aimed based on the training of NN. Actually, many industrial process require a classification task, this activity demands too much time, it converts in an expensive and inefficient procedure. It is presented requirements for the data processing and the implement of several elements such as sensors and electric motors. Therefore, mechanical elements and manufacture to elaborate automated devices and vision systems. It is showed a device to classify mechanical elements by image processing using artificial intelligence and the programming of a microcontroller, which consists in a transporter band, a digital camera and a rotating container. Using these elements, the images are classified for this particular case images of screws. The rotation container is synchronized with the digital camera for its later angular movement. The angles are constant according to the class of elements to stock, such screws classification show mean errors of 0.3%.

#### Automatization, Classification, Neural Networks

**Citación:** PERÉZ-CHIMAL, Rosa Janette, LUNA-PUENTE, Rafael, GUERRERO-ORDAZ, Salvador y JAMAICA-GONZÁLEZ, Alejandro. Clasificador automático de elementos mecánicos empleando Redes Neuronales. Revista de Tecnología e Innovación. 2018, 5-16: 33-41.

\*Correspondencia al Autor (Correo Electrónico: rperez@utsalamanca.edu.mx)

† Investigador contribuyendo como primer autor.

## Introducción

Muchos procesos industriales requieren de soluciones para hacer eficiente la separación de piezas, lo que implica menores tiempos, menores costos de producción y mayor calidad. En muchas empresas necesitan de procesos de clasificación para optimizar la calidad de sus productos.

Actualmente existen diversos mecanismos para la clasificación de formas geométricas, el reto ahora es la clasificación de formas no geométricas.

Por ejemplo, en la industria alimenticia se clasifican frutas y verduras en función de su forma, color y estado de descomposición. Específicamente en este tipo de empresas es indispensable que el proceso de clasificación sea de forma limpia y no invasiva.

En otras áreas como la industria automotriz o la elaboración de piezas metálicas como los tornillos, es necesaria la precisión, la velocidad de clasificación y el monitoreo remoto de los procesos.

Muchas de las piezas que se realizan ya no son en serie, debido a que el consumismo actual solicita cada vez más productos únicos, por lo que se fabrican piezas sobre pedido. Al mismo tiempo deben de fabricarse de forma rápida y en la misma máquina para distintos diseños con el objeto de ahorro de recursos.

Lo anterior solo podría darse por medio de personas que se encontrarán en planta para, de manera manual, monitorear y colocar las piezas en el lugar indicado. Sin embargo, y como todos los procesos donde interviene el hombre, diversos factores personales y ambientales puede modificar la calidad del proceso de clasificación.

Actualmente se pueden combinar los procesos de clasificación de forma limpia, no invasiva, y con las propiedades de inteligencia, que hasta hace poco, solo los humanos podrían proporcionar [1].

Las redes neuronales (ANN) ofrecen un modelo de inteligencia artificial el cual proporciona, por medio de un entrenamiento, la calidad que se requiere para hacer estas labores.

Con el fin de automatizar este tipo de procesos de clasificación en la industria se presenta una opción de carácter aplicable el cual consiste en

## Marco Teórico

### Redes Neuronales Artificiales

La inteligencia artificial parte del estudio del cerebro y del sistema nervioso humano. Lo que se busca es programar redes neuronales artificiales para un procesamiento automático de información por medio del entrenamiento de estas.

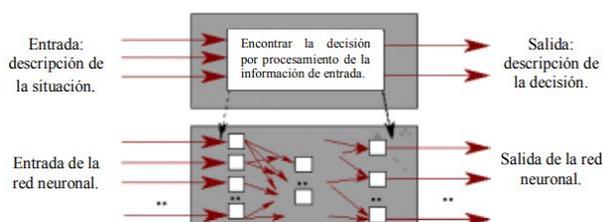
El cerebro está compuesto de neuronas las cuales se comportan de forma individual para el procesamiento de información, esta información viaja a través de las mismas basado en la organización de los conectores neuronales [2].

Una red neuronal artificial consiste en una interconexión de neuronas que interactúan entre sí para producir un estímulo. Esta se compone de neuronas las cuales reciben una serie de instrucciones para emitir una salida específica.

Las redes neuronales tienen ciertas características que las distinguen:

- Las redes neuronales pueden ser entrenadas, para en algún momento dado estas puedan reaccionar a un estímulo no programado y poder trabajar con el reconocimiento de patrones.
- Las redes neuronales aprenden la experiencia, en otras palabras, pueden modificar su comportamiento como respuesta a su entorno.
- Pueden generalizar de ejemplos anteriores para ser insensible a pequeños cambios [3].

Las propiedades de las redes neuronales, sirven para crear máquinas capaces de resolver problemas los cuales eran difíciles de manejar por computadoras, hasta hace poco solamente a humano se le podía caracterizar por su inteligencia, ahora estas máquinas ofrecen, si bien no la inteligencia total del humano, si una inteligencia parcial para la solución de problemas específicos [4].



**Figura 1** Esquema del proceso de decisión [4]

La combinación de máquinas e inteligencia artificial abre un mundo de posibilidades en el ámbito industrial. Lo anterior permite realizar procesos de forma más limpia y segura para el ser humano.

### Procesamiento de datos

Las redes neuronales se trabajan mediante algoritmos los cuales reciben instrucciones específicas para entrenar a las neuronas. Estos pueden ser trabajados en software como MatLab el cual ofrece grandes ventajas como, programación, procesamiento de datos y comunicación con otras plataformas.

Las principales características de Matlab son:

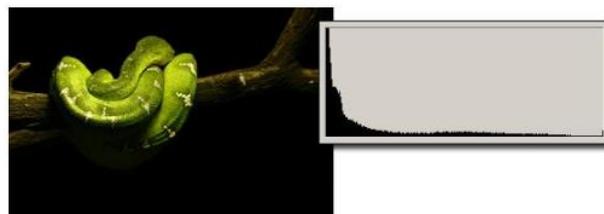
- Es un lenguaje de alto nivel, lo que lo hace funcional para el cálculo numérico, la visualización y el desarrollo de aplicaciones.
- Cuenta con varias herramientas que estimula un entorno interactivo para el diseño y la solución de problemas.
- Posee una gran cantidad de funciones matemáticas para álgebra lineal, estadística, análisis de Fourier, filtrado, optimización, integración numérica y resolución de ecuaciones diferenciales ordinarias, por mencionar algunas.
- Contiene herramientas para creación de gráficos integrados.
- Tiene herramientas de desarrollo para el mantenimiento del código, así como para maximizar el rendimiento.
- Herramientas para crear aplicaciones con interfaces gráficas personalizadas.
- Funciones para integrar algoritmos basados en Matlab con aplicaciones y lenguajes externos tales como C, Java, .NET entre otros.

Matlab es muy utilizado para el procesamiento de imágenes debido a su gran capacidad y su versatilidad en el procesamiento de las mismas, Matlab almacena las imágenes como matrices, donde cada elemento de la matriz corresponde a un pixel, por lo que su manipulación es sumamente exacta.

Al digitalizar una imagen en Matlab, se divide en filas y columnas, a la intersección de estas dos se le conoce como pixel por lo que cada pixel tiene entonces, una coordenada en la matriz. Debido a que las imágenes son representadas de forma numérica estas pueden ser alteradas conforme uno lo desee. Las imágenes en blanco y negro son más fácilmente manipulables debido a que se cuenta con solo un solo canal o “capa” de trabajo. Matlab ofrece el cambio de entre una imagen a color, escala de grises y blanco y negro con instrucciones simples [5].

Para muchas aplicaciones de procesamiento de imágenes se utiliza un Histograma, el cual es una gráfica que representa los niveles de intensidad del color de la imagen respecto al número de pixeles de la misma. Este se da con el comando *imhist* el comando que nos muestra una gráfica donde el eje horizontal representa los diferentes tonos de gris desde el negro hasta el blanco. Y en el eje vertical el número de pixeles que contiene cada imagen del tono de grises.

En la figura 2 se observa como la mayor cantidad de pixeles se encuentran en el lado izquierdo del histograma donde se localiza el negro.



**Figura 2** Histograma de imagen con muchas zonas oscuras [5]

Esta herramienta sirve para poder tener información de una imagen sin importar la posición de la misma.

También se pueden utilizar otras herramientas como [6]:

Tipo	Sintaxis y parámetros
'average'	>> Fspecial ('average', [r c]) Filtro de la media $r \times c$ . Por defecto, aplica el filtro $3 \times 3$ . Si ponemos un solo número $r$ indica que es una máscara cuadrada $r \times r$ .
'gaussian'	>> Fspecial ('gaussian', [r c], sigma) Filtro gaussiano $r \times c$ con desviación típica sigma. Por defecto, aplica filtro gaussiano $3 \times 3$ con $\sigma=0,5$ .
'prewitt'	>> Fspecial ('prewitt') Filtro gradiente Prewitt $3 \times 3$ . Devuelve la máscara $mx$ que aplica el gradiente vertical. La máscara que aplica el gradiente horizontal se obtiene con la traspuesta de la anterior, es decir, >> $my=mx'$ .
'sobel'	>> Fspecial ('sobel') Filtro gradiente Sobel $3 \times 3$ . Devuelve la máscara $mx$ que aplica el gradiente vertical. La máscara que aplica el gradiente horizontal se obtiene con la traspuesta de la anterior, es decir, >> $my=mx'$ .
'laplacian'	>> Fspecial ('laplacian', alpha) Filtro laplaciano $3 \times 3$ cuya forma viene dada por alpha (valor entre 0 y 1). Por defecto, $\alpha = 0,5$ .

**Tabla 1** Máscaras para el procesamiento de imágenes

Matlab servirá entonces para el procesamiento de imágenes que se tomaran de las piezas a utilizar bajo estos comandos.

### Automatización Industrial

En la automatización se busca proporcionar máquinas capaces de percibir información de objetos; posiciones, colores, tamaños, distancias para su manipulación. Automatizar implica reducir en gran medida la intervención humana en cualquier proceso, remplazando a este con máquinas capaces de operar de la misma o mejor manera que el hombre.

Antes de la intervención de las máquinas a las empresas, la clasificación de piezas u objetos se realizaba completamente de forma manual por el hombre. Posteriormente con la intersección de los motores eléctricos en la industria se facilita a los trabajadores con bandas transportadoras la manipulación del trabajo.

Desde este momento se utilizaron gran cantidad de componente para la manipulación de los motores como; variadores de velocidad. Posteriormente se anexan distintos elementos electrónicos como sensores para elaborar maquinaria más completa que haga de manera autónoma muchos de los trabajos que se realizaban hasta entonces por trabajadores.

Si bien la automatización no solo tiene como objetivo volver al hombre obsoleto en procesos industriales, sino más bien facilitar el trabajo de este, además de reducir esfuerzos y riesgos. Dando una solución en ambientes nocivos, muy precisos o tareas inalcanzables para el hombre.

Una de las máquinas que han facilitado y sobre todo, generado muchas ganancias a la industria, son las máquinas clasificadoras.

Las máquinas clasificadoras pueden ser diseñadas de manera mecánica, como las clasificadoras de tamaños por medio de de bandejas de distintos tamaños y rodillos los cuales hacen que las piezas se desplacen sobre estas y caigan en el orificio apto para su tamaño [7].

Estas máquinas no han funcionado del todo bien, debido a que al manejar productos sensibles o frágiles, la intervención humana es mejor opción. Es por esto que se han seguido innovando en técnicas para la construcción de éste tipo de máquinas [8].

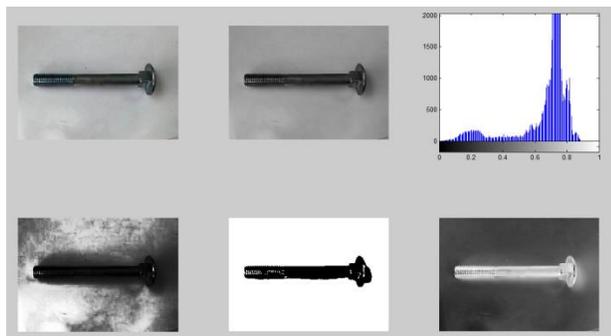
### Desarrollo

En este proyecto se utiliza la inteligencia artificial para clasificar una serie de piezas por medio de procesamiento de imágenes y programación en un microcontrolador.

El objetivo general es construir un prototipo que permita realizar pruebas con una banda transportadora, una cámara y un contenedor giratorio. Realizando de esta manera la clasificación de figuras, para este caso tornillos, mediante la plataforma giratoria sincronizada con una cámara que realice las capturas, y que pueda posicionarse en un ángulo concreto con cierta precisión para que almacene los tornillos.

### Entrenamiento de la red

Para el entrenamiento de la red se realiza la captura de 50 imágenes del mismo tornillo en diferentes posiciones, se tomaron 5 tornillos diferentes como referencia. Primero se quita el fondo para obtener una imagen sin fondo.



**Figura 3** Resultados obtenidos del procesamiento de imágenes

En total se obtuvieron 250 perfiles a los cuales se les aplicó las propiedades antes mencionadas para tener como resultado la imagen del tornillo en un fondo completamente blanco (figura 4) para obtener las características de éste y proceder a realizar la base de datos con cada una de ellas y llevar a cabo el entrenamiento de la red neuronal.



**Figura 4** Imágenes procesadas de dos tornillos

Una vez elaborada la base de datos con las características de los cinco tornillos se inició el desarrollo y entrenamiento de la red neuronal, para que una vez capturada la imagen y procesada, la red neuronal nos arrojará el tipo de tornillo con el que se estaba manejando en el momento y poderle mandar la orden a la parte mecánica del proyecto.

```

clc, clear all, close all
%%
I=double(imread('tor5_41.jpg'));

R(:,:,1)=I(:,:,1);
G(:,:,2)=I(:,:,2);
B(:,:,3)=I(:,:,3);
for i=1:326
    W5_118 ((i-1)*484+1:484*i, 1)=R(i, :);
end
for i=1:326
    W5_119 ((i-1)*484+1:484*i, 1)=G(i, :);
end
for i=1:326
    W5_120 ((i-1)*484+1:484*i, 1)=B(i, :);
end
save datrgb5_41 W5_118 W5_119 W5_120
RES=[];
ST=[1 0 0 0;...
    0 1 0 0;...
    0 0 1 0;...
    0 0 0 1;...
    0 0 0 1];
ST=ST;
for t=1:5
    for d=1:50
        RES(:,d+50*(t-1))=ST(:,t);
    end
end

```

**Figura 5** Programa para leer los datos de la imagen

ISSN: 2410-3993

ECORFAN® Todos los derechos reservados

Para continuar se obtendrán los valores conocidos como: mean, skewness, kurtosis y std para así con estos valores comparar cada una y diferenciar cada una. Después de obtener los datos estos se ordenarán de manera similar a una matriz, de esta manera al acomodar los datos generamos una salida para mandar la señal al Arduino (figura 5).

Una vez almacenado los datos de cada una de las 250 fotos se crea la red neuronal por medio del siguiente código.

```

clc, clear all, close all
load matriz-473367.mat %se carga la matriz
load salida.mat %y salida

I= W(473352:473367,:);%('matriz-473367.mat')
Y= RES;%('salida.mat')
rango=minmax(I);
net=newff(rango,[555555],{'logsig','logsig','logsig','logsig','logsig','logsig'});
salida1= sim(net,I); %se genera la salida
plot(salida1,'o');
hold on; %se mantiene la red abierta
plot(Y,'+r');

%%
% con entrenamiento
net= train(net,I,Y); %con este comando se comienza el entrenamiento
salida2= sim(net,I);
mesh(salida2);
hold on;
plot(Y,'+r');

net.Iw{1,1}
net.Lw{2,1}
valor=mse(net,I,Y);

```

**Figura 6** Código para el entrenamiento de la red

Se realiza la combinación de varios programas anteriores para la realización de la comparación de datos de fotografías, tales como el cambio de color, quitar el fondo, y sacar los datos principal mente para comparar el archivo net que se obtuvo de la matriz ya entrenada. Una vez procesada cierta fotografía se guardan los datos obtenidos de la imagen y se carga el archivo creado a partir de la red neuronal, una vez cargado el archivo se crea una variable salida para comparar los datos que se tienen en la red neuronal con los de la foto del programa anterior, de esta manera el programa proporciona una respuesta específica sobre cuál es el tornillo que se fotografió proporcionando un 1 en el número de posición al cual se le asignó ese tornillo.

## Control

Para la realización del control del proceso se utiliza un microcontrolador Arduino como tarjeta de adquisición de datos. Para esto es necesario preparar la tarjeta Arduino para hacer conexión con Matlab y la etapa de potencia.

Por medio de un sensor de presencia, se detecta el momento en el que el tornillo pasa por debajo de la cámara en la posición ideal para la captura de la imagen. Adicional a esto se cuenta con un botón de arranque y paro en caso de ser necesario poder detener la banda.

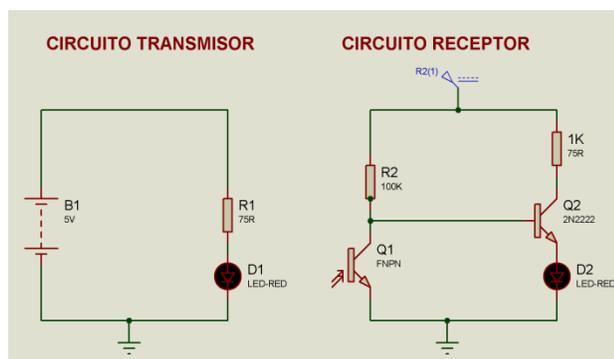


Figura 7 Diagrama eléctrico del sensor

Desde Matlab se manda llamar a la captura de la imagen por medio de la cámara instalada. Esta imagen se procesará por medio del programa en Matlab con la red neuronal, el programa hará la comparación y mandará una salida que se procesará en el Arduino para identificar la posición de la bandeja giratoria donde caerá el tornillo.

Para interactuar con la tarjeta, desde el entorno de Matlab; se crea un objeto para acceder a la información de puerto de comunicación a la que está conectada la tarjeta. Se hace con la línea de comando, de la siguiente forma:

```
cIO = arduino('COM3')
```

Para la manipulación de la banda transportadora se tomó en cuenta que esta es arrastrada por la fricción de los tambores, que a la vez este es accionado por su motor. Esta fricción es la resultante de la aplicación de una tensión a la banda transportadora. El otro tambor suele girar libre, sin ningún tipo de accionamiento, y su función es servir de retorno a la banda. La banda es soportada por rodillos entre los dos tambores. Denominados rodillos de soporte.

El motor utilizado para el prototipo es un servomotor GWS S35/STD. Este servomotor cuenta con una rotación continua a comparación de otros, se alimenta de 4.8V – 7.5V máximo y el voltaje que nos proporciona la placa Arduino es de 5V.

El código utilizado para el posicionamiento de la bandeja giratoria es el siguiente:

```
#include <Servo.h>
Servo myservo;
int divisor=4;
int sentido=1;

void setup()
{
  myservo.attach(9);
}
void loop() {
  myservo.write(0);
  delay(10000);
  myservo.write(180/divisor); //indica que vaya a 45°, 90°, ...
  delay(10000);
  if(sentido==1)
  {
    divisor=divisor/2;
    if(divisor==1)
    {
      Sentido=0;
    }
  }
  else
  {
    divisor=divisor*2;
    if (divisor==4);
    {
      Sentido=1;
    }
  }
}
```

Figura 8 Programa para el posicionamiento de la bandeja

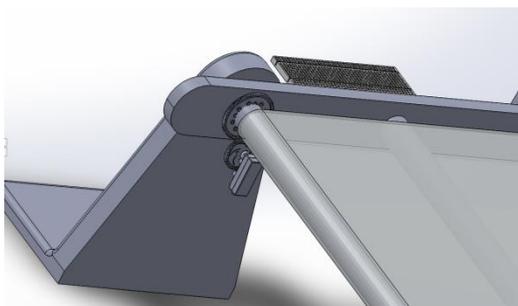
La placa tiene tres tipos de alimentación: a través de los pines de entrada, por USB o por alimentación externa. El pin de entrada de la placa Arduino en este caso se utiliza solamente para conectar GND de la placa Arduino y el negativo de la alimentación del motor.

El conector USB (tipo B) que conecta con el ordenador. A través de él se programa la placa y también alimenta el circuito. Para la entrada de corriente al servomotor, se adapta un cable USB por el conector tipo B para poder conectarlo a la placa Protoboard, y el otro extremo tipo A se conecta al ordenador para obtener la corriente del servomotor fácilmente sin depender de una fuente de alimentación externa. Arduino también dispone de pines de salida a 5 V que facilitarían el conexionado, pero cuando el servomotor se pone en funcionamiento la placa necesita más potencia de la que ofrece el puerto USB y se reinicia.

### Diseño y construcción

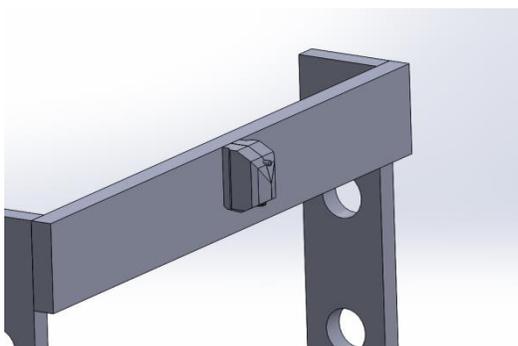
El diseño consiste en una banda transportadora compuesta de una estructura metálica, motor eléctrico, engranajes, rodillos de acrílico y banda de lona.

Los engranajes sirven para conectar el motor con la banda transportadora y así controlar su movimiento (ver figura 8).



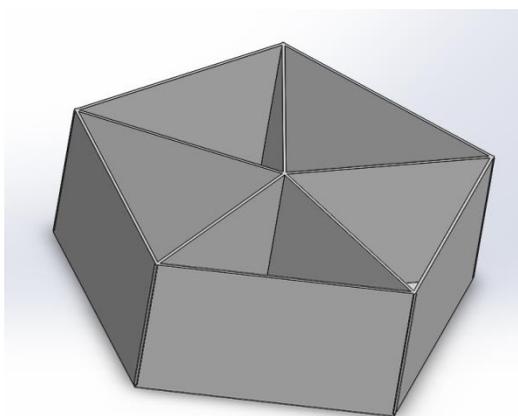
**Figura 9** Diseño mecánico

A mitad de la banda se encuentra un arco metálico con base para la cámara, la cual tomará la foto del tornillo al pasar por ella. Se encuentra a un costado una base para el circuito eléctrico que controla todo el sistema y los botones de paro y arranque.



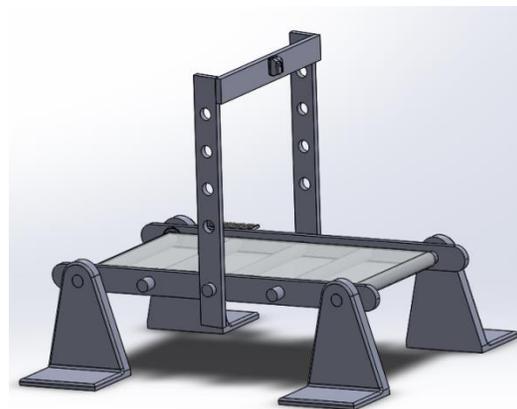
**Figura 10** Colocación de la cámara para la captura de las fotografías de los tornillos

La base contenedora de los tornillos es una plataforma giratoria con 5 separaciones que por medio de un motor se coloca en la posición indicada programada dependiente del tornillo deseado.



**Figura 11** Bandeja giratoria contenedora de tornillos

La estructura final se muestra en la figura 11 donde se puede observar a escala la banda transportadora y la colocación de la cámara. A esto se le conectará la PC para hacer el procesamiento de imágenes y la circuitería para el control de los motores y del arranque y paro.



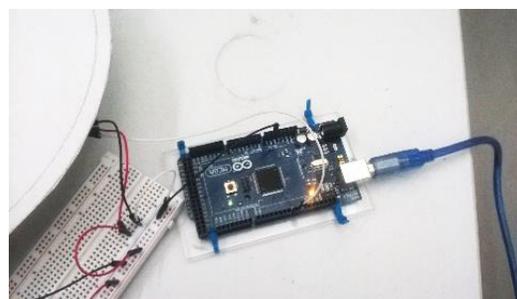
**Figura 12** Estructura

## Resultados

Como resultado se presenta una banda transportadora con clasificador automático de tornillos. El cual funciona por medio de sensores de presencia los cuales detectan una pieza y mandan la señal a la PC la cual por medio de programación indica a la cámara que debe tomar la captura de una fotografía.

Una vez capturada esta imagen se procesa por medio del entrenamiento de la red neuronal y se identifica en función a la base de datos el tipo de tornillo. Una vez determinado esto se manda la señal al Arduino el cual posicionará el motor que a su vez moverá la bandeja giratoria en el lugar asignado para ese tornillo.

Tras el montaje de la base giratoria bajo el control de la placa Arduino se llevan a cabo dos estudios diferentes para comprobar el buen funcionamiento del sistema.



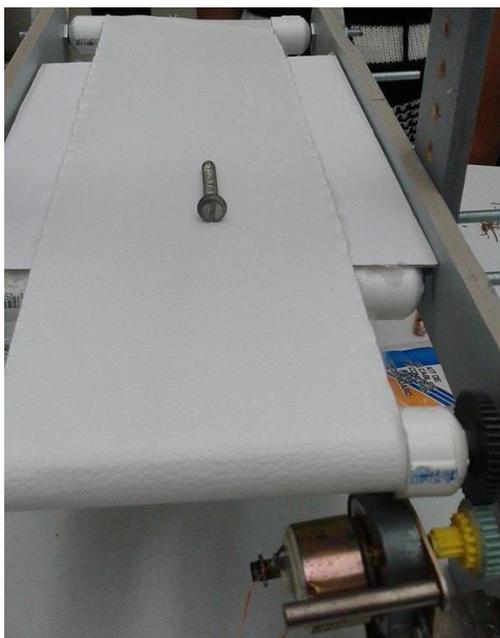
**Figura 13** Pruebas para la base giratoria

En un primer estudio, el propósito de la prueba de repetitividad es determinar la exactitud de la repetición angular exigida a la base por la placa controladora.

El motivo de la siguiente prueba, la de fiabilidad, es comprobar la precisión del ángulo de giro configurado. De esta manera también se comprueba si el servo cumple las especificaciones y si Arduino tiene un buen funcionamiento.

En la configuración de Arduino se ha opta por hacer girar el servomotor con la siguiente secuencia para así provocar el movimiento del servo a la posición de 0° desde distintas posiciones:

0° → 45° → 0° → 90° → 0° → 180° → 0° → 90° → 0° → 45°...



**Figura 14** Pruebas

Se realizan pruebas con éxito, teniendo como observaciones el control de velocidad de la banda la cual debe tener una velocidad idónea para la captura de la imagen y el tiempo de posicionamiento de la bandeja. El producto final se observa en la figura 15.



**Figura 15** Prototipo de clasificadora

## Conclusiones

La programación en conjunta con distintas plataformas como Arduino y Matlab abren un sin fin de aplicaciones para la automatización de diversos proyectos. Al observar la creación y entrenamiento de una red neuronal se vio que con esta se puede crear una inteligencia artificial que responda a ciertos patrones como a preguntas y dudas.

Si bien la clasificación de piezas como tornillos en una poderosa herramienta para la industria automotriz, se hace énfasis que se eligió la separación de tornillos por la facilidad de establecer características marcadas para efectuar la selección de estos por diferentes métodos. Sin embargo la aplicación finalmente obtenida se puede hacer extensiva a procesos de separación de gran variedad de productos con sus diferentes características. Con el fin de buscar una solución al problema planteado se propone realizar el diseño de una máquina que clasifique y separe tornillos de acuerdo a su tamaño, para que los operarios sean más eficientes y mejore la producción.

Otras aplicaciones igualmente potenciales pueden ser en la industria alimentaria para la selección de alimentos como frutas y verduras dependiendo del tamaño y forma del mismo. Adecuando esta máquina se puede hacer adicional a la implementación dada una clasificadora de color para detectar fruta en malas condiciones.

Otra aplicación de este dispositivo puede ser en un área distinta a la industrial. En el área médica o biológica puede ser aplicado el principio del método de procesamiento de imágenes se pueden detectar afectaciones en la piel de personas o plantas para detección de enfermedades. Las cuales en función a imágenes ya afectadas se realiza una comparación con la que se está probando y se separa tejido sano del no sano.

El principio de los clasificadores es el mismo, las aplicaciones son infinitas.

### Referencias

- [1] R. Mier Pérez, J. L. García de Ceca, M. R. Díez Barra y Fernández Golfin S, «Aplicación de redes neuronales a la clasificación de madera estructural,» Congresos Forestales, nº 16378, 2007.
- [2] F. R. Bello Acosta, «Nuevo enfoque en el diseño y entrenamiento de redes neuronales para la clasificación,» Universidad Autónoma de Nuevo León, San Nicolás de los Garza, 2001.
- [3] S. Ledesma , «Las Redes Neuronales: implementación y consideraciones prácticas,» MICAI, vol. 6, pp. 1-50, 2006.
- [4] D. J. Matich, «Informática Aplicada a la Ingeniería de Procesos – Orientación I,» Universidad Tecnológica Nacional, Rosario, 2001.
- [5] Manna Andrea, «Introducción al procesamiento de imágenes con Matlab 1era parte,» Universidad de Buenos Aires , Buenos Aires, 2016.
- [6] J. G. Velásquez Aguilar, U. Alcázar Carreño, R. Mañón Abarca, L. Martínez Rebollar y G. Ortíz Ojeda, «Herramienta de apoyo para cursos de procesamiento digital de imagen,» de Congreso Internacional de Cómputo en optimización y software, 2006.
- [7] M. F. Angos Mediavilla y H. A. Calvopiña Enriquez, «Diseño, construcción y simulación de una máquina clasificadora de frutos por su tamaño,» Universidad de las fuerzas armadas, Sangolquí, 2013.
- [8] A. Gómez C y C. Gómez E, «Diseño de una máquina clasificadora y limpiadora de vegetales,» Universidad Central de Venezuela, Caracas, 2005.