

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**IMPLEMENTACIÓN DE UN SISTEMA DE DETECCIÓN DE
MONEDAS POR VISIÓN ARTIFICIAL**

**SISTEMA DE DETECCIÓN DE PATRONES SIMPLES DE
MONEDAS**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN ELECTROMECAÁNICA**

ALEXANDER DAVID ANDRANGO AYO

DIRECTORA: ARACELY INÉS YANDÚN TORRES

DMQ, septiembre 2022

CERTIFICACIONES

Yo, Alexander David Andrango Ayo declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



ALEXANDER DAVID ANDRANGO AYO

alexander.andrango@epn.edu.ec

alexanderandrangoayo@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Alexander David Andrango Ayo, bajo mi supervisión.



ARACELY INÉS YANDÚN TORRES

DIRECTOR

aracely.yandun@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmo que el trabajo de integración curricular aquí descrito, así como el producto resultante del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

ALEXANDER DAVID ANDRANGO AYO

DEDICATORIA

Este trabajo realizado con esfuerzo y dedicación lo dedico a las personas que más aprecio.

AGRADECIMIENTO

A Dios y a mis padres por el amor, apoyo y comprensión que desde el principio me dieron, brindándome muchos consejos y la perseverancia para alcanzar las metas planteadas en mi vida.

A la Escuela Politécnica Nacional que por medio de todos los ingenieros e ingenieras me impartieron su conocimiento para aprender cosas nuevas y ser un excelente profesional.

A toda mi familia que estuvo pendiente en el progreso de mi formación profesional.

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN.....	VII
ABSTRACT.....	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general.....	1
1.2 Objetivos específicos.....	2
1.3 Alcance	2
1.4 Marco teórico	2
Visión Artificial	2
Aplicaciones de la Visión Artificial	3
Python	3
Librería OpenCV	4
Librería Numpy.....	5
Dólar estadounidense	5
Cámara Web	7
Sistemas de Iluminación	7
Proceso de la Visión Artificial.....	9
2 METODOLOGÍA.....	11
2.1 Requerimientos mínimos del sistema	11
Cámara web.....	11
Software de programación Pycharm	12
Espacio de trabajo	13
2.2 Pruebas de cámara.....	13

2.3	Diseño del Algoritmo.....	14
	Desarrollo del algoritmo	14
3	RESULTADOS.....	21
3.1	Moneda de 1 dólar.....	21
3.2	Moneda de 50 centavos	22
3.3	Moneda de 25 centavos	23
3.4	Moneda de 10 centavos	24
3.5	Moneda de 5 centavos	25
3.6	Moneda de 1 centavo	26
3.7	Monedas de otros países	27
4	Conclusiones.....	29
5	Recomendaciones	30
6	REFERENCIAS BIBLIOGRÁFICAS	31
7	ANEXOS.....	33
	ANEXO I.....	33
	ANEXO II.....	35
	ANEXO III.....	36
	ANEXO IV.....	41
	ANEXO V.....	42

RESUMEN

En el capítulo 1 se describen los objetivos, la información básica y los procesos necesarios para la implementación de un sistema de visión artificial, en el caso específico de este trabajo, un sistema para para la detección de monedas estadounidenses.

En el capítulo 2 se detalla la metodología del sistema, en este caso la determinación de los requerimientos mínimos, mismos que son: la cámara web, el espacio de trabajo, y el IDE con el que se trabaja. Adicionalmente, en este capítulo se detalla la lógica de programación del algoritmo de visión artificial implementado.

En el capítulo 3 se realizan las pruebas de funcionamiento con cada una de las monedas estadounidenses y la comprobación de su detección en el sistema, esta detección de monedas es realizada a través del área de las mismas. Además, se implementó en el sistema la visualización de un mensaje al momento de colocar monedas de otros países, mismas que tienen diferentes áreas que las monedas estadounidenses.

En el capítulo 4 y 5 se desarrolla las conclusiones y recomendaciones de la implementación del sistema de detección de monedas estadounidenses.

PALABRAS CLAVE: visión artificial, IDE Pycharm, cámara web, monedas estadounidenses, detección.

ABSTRACT

The chapter 1 describes the objectives, the basic information and the necessary processes for the implementation of an artificial vision system, in the specific case of this work, a system for the detection of US coins.

The chapter 2 details the methodology of the system, in this case the determination of the minimum requirements, which are: the webcam, the workspace, and the IDE with which it works. Additionally, this chapter details the programming logic of the implemented artificial vision algorithm.

The chapter 3, the performance tests are carried out with each of the US coins and the verification of their detection in the system, this detection of coins is carried out through the area of the same. In addition, the display of a message was implemented in the system when placing coins from other countries, which have different areas than US coins.

The chapter 4 and 5 the conclusions and recommendations of the implementation of the US currency detection system are developed.

KEYWORDS: computer vision, Pycharm IDE, webcam, US coins, detection.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Para proceder con el desarrollo del siguiente trabajo, en este caso de la implementación de un sistema de detección de monedas estadounidenses por visión artificial, como primer paso se realiza la investigación básica necesaria sobre los temas que abarca la visión artificial, así como de los diferentes procesos que existen para implementar el sistema. Luego, se determina los requerimientos mínimos para que funcione el mismo, en este caso los elementos son: la cámara web, el espacio de trabajo y el software de programación, el cual se va a encargar de compilar el programa realizado. Para el sistema implementado se usa el software IDE 'Pycharm', mismo que utiliza el lenguaje de programación Python, la librería OpenCv, que se usa específicamente para visión artificial y la librería Numpy, que sirve para realizar matrices con mayor velocidad y guardar gran cantidad de información. Después, se desarrolla el algoritmo para que funcione el sistema, teniendo en cuenta los procesos de transformación de la imagen de la cámara web a datos binarios y considerando que, al momento de realizar el programa, este tenga un orden correspondiente y sea entendible. La lógica de programación del algoritmo del sistema de detección de monedas estadounidenses es a través del área, donde su unidad de medida son los pixeles cuadrados, a través de estos datos el algoritmo desarrollado pueda leer la información que este necesita y detecte correctamente. Además, en el algoritmo, se implementa la opción de detección de monedas de otros países, mismas que tienen áreas diferentes a las monedas estadounidenses. Posteriormente, se realiza 10 pruebas de funcionamiento del sistema implementado a cada una de las monedas estadounidenses para comprobar su detección y que el valor de cada moneda se visualice en la imagen final. Además, como se implementa una forma de detección de monedas de otros países se realiza varias pruebas de funcionamiento y en el sistema se visualizará como 'Moneda Desconocida' obteniendo los resultados esperados que se detallarán en la sección de análisis de resultados. Finalmente, al terminar la implementación del sistema de detección de monedas estadounidenses por visión artificial se detallan las conclusiones y recomendaciones.

1.1 Objetivo general

Implementar un sistema de detección de monedas estadounidenses por visión artificial

1.2 Objetivos específicos

Para el desarrollo del proyecto se establecieron los siguientes objetivos específicos:

1. Determinar los requerimientos mínimos para el sistema
2. Desarrollar el algoritmo necesario para el reconocimiento de monedas estadounidenses
3. Realizar las respectivas pruebas de funcionamiento.

1.3 Alcance

Este proyecto se basa en la implementación de un sistema de detección de monedas estadounidenses por medio de patrones que identifiquen las mismas. Para esto, haciendo uso de una cámara web, y a través del lenguaje de programación "Python", se implementará el algoritmo necesario para la respectiva identificación de monedas estadounidenses. Se realizarán pruebas de funcionamiento del sistema implementado con monedas de 1 centavo, 5 centavos, 10 centavos, 25 centavos, 50 centavos y 1 dólar.

1.4 Marco teórico

En el transcurso de los años la tecnología ha ido avanzando considerablemente en la automatización de procesos. Uno de esos avances es el uso de la visión artificial para la detección de patrones simples y en el siguiente subcapítulo se detallará los elementos básicos que conforman el presente proyecto.

Visión Artificial

La visión artificial es un campo que abarca varios procesos como son: capturar, analizar y procesar imágenes del mundo con la finalidad de proporcionar información que pueda ser procesada en sistemas automatizados [1].

Además, hoy en día la visión artificial utiliza las más recientes tecnologías referentes a inteligencia artificial para proveer a equipos industriales de ver y analizar las tareas con la finalidad de tomar decisiones rápidas [2].

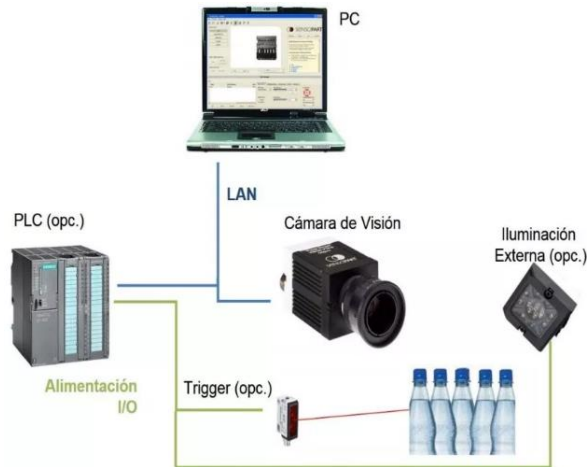


Figura 1.1. Sistema automatizado por visión artificial en un proceso industrial [1]

Aplicaciones de la Visión Artificial

La visión artificial es una parte muy importante en la logística, la fabricación y las operaciones inteligentes en varios procesos. Las cámaras de visión artificial, las computadoras industriales y los sensores IoT integrados, aportan análisis, eficiencia e inteligencia en cada momento del proceso de fabricación [2]. Sus aplicaciones más típicas son utilizadas para clasificación, control de calidad, conteo de productos, control de rotación, posicionamiento y “pick and place” [1].



Figura 1.2. Control de calidad por medio de visión artificial [3]

Python

Es un lenguaje de programación orientado a objetos (de alto nivel), que se utiliza principalmente para desarrollar sitios web y aplicaciones informáticas. “Python” es un lenguaje relativamente simple, por lo que su aprendizaje es fácil, por lo tanto, sus

programadores pueden traducir y leer el código desarrollado más fácilmente que en otros lenguajes [4].

“Python” es un lenguaje de programación multiplataforma, el cual permite desarrollar fácilmente programas en cualquier sistema operativo. Dentro de las principales aplicaciones se tiene: análisis de datos y “big data”, minería de datos, data science, inteligencia artificial, blockchain, aprendizaje automático, desarrollo web, juegos y gráficos 3D [5].

```
x = int(Momentos["m10"]/Momentos["m00"])
y = int(Momentos["m01"] / Momentos["m00"])

if area < 16550 and area > 13500:
    font = cv2.FONT_HERSHEY_SIMPLEX
    cv2.putText(imagen_A6, '50 ctvs', (x, y), font, 0.75, (0, 255, 0), 2)
    suma = suma + 0.5

elif area < 12600 and area > 11000:
    font = cv2.FONT_HERSHEY_SIMPLEX
    cv2.putText(imagen_A6, '1$', (x, y), font, 0.75, (0, 255, 0), 2)
    suma2 = suma2 + 1.0
```

Figura 1.3. Lenguaje de programación “Python”

Librería OpenCV

OpenCV es una librería de computación artificial desarrollada por Intel mayormente utilizado para el desarrollo de algoritmos para visión artificial. Las diferentes aplicaciones de esta biblioteca incluyen: la robótica, el procesamiento de videos o imágenes, seguimiento y reconocimiento de objetos, reconocimiento y detección de rostros, análisis de formas, reconocimiento de placas de vehículos, realidad aumentada, reconstrucción 3D y otras más aplicaciones [6].

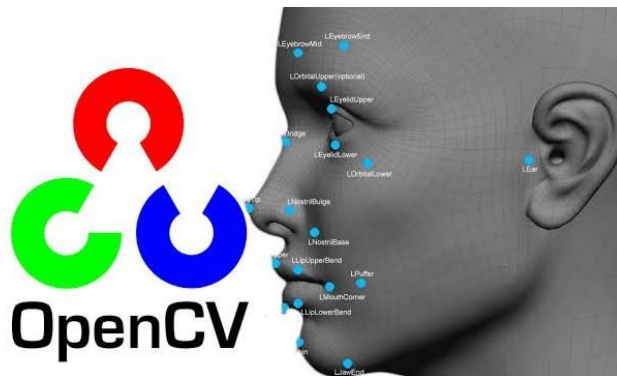


Figura 1.4. Librería OpenCV para visión artificial [7]

Librería Numpy

Numpy es una librería de “Python” especializada en el análisis de datos y el cálculo numérico, desarrollada para gran cantidad de datos. Incluye una nueva clase de elementos llamados matrices o “arrays” que permite representar funciones y colecciones de datos de igual patrón en varias dimensiones.

La principal ventaja de esta librería es el procesamiento, ya que es mucho más veloz que las mismas listas que tiene “Python”, y la hace ideal para procesamiento de matrices y vectores de extensas dimensiones [8].

```
>>> # Array de una dimensión
>>> a1 = np.array([1, 2, 3])
>>> print(a1)
[1 2 3]
>>> # Array de dos dimensiones
>>> a2 = np.array([[1, 2, 3], [4, 5, 6]])
>>> print(a2)
[[1 2 3]
 [4 5 6]]
>>> # Array de tres dimensiones
>>> a3 = np.array([[[1, 2, 3], [4, 5, 6]], [[7, 8, 9], [10, 11, 12]]])
>>> print(a3)
[[[ 1  2  3]
 [ 4  5  6]]

 [[ 7  8  9]
 [10 11 12]]]
```

Figura 1.5. Matrices realizadas con librería Numpy [8]

Dólar estadounidense

El dólar es la moneda oficial de los Estados Unidos de Norteamérica, que si bien es el lugar donde se emite, también es usado en ocho países que lo tienen como moneda oficial, entre ellos: Estados Unidos, Islas Marshall, El Salvador, Palaos, Timor Oriental, Micronesia, Zimbabwe y Ecuador [9].

La Tabla 1.1 detalla las monedas de dólar estadounidense. Acorde a esta tabla, se puede identificar la denominación de cada una de las monedas, su dimensión en milímetros, así como su imagen respectiva en cada uno de sus lados.

Tabla 1.1. Monedas estadounidenses

Denominación	Tamaño	Imagen
1 centavo	19 mm	
5 centavos	21,5 mm	
10 centavos	18 mm	
25 centavos	24,2 mm	
50 centavos	30,6 mm	
1 dólar	26,5 mm	

Cámara Web

La cámara es el elemento principal de los sistemas basados en visión artificial ya que por medio de este hardware se capturará la imagen con la cual se trabajará. Las principales partes de una cámara web son: lente óptico, micrófono, base y puerto de conexión [10].



Figura 1.6. Cámara web [10]

Sistemas de Iluminación

El sistema de iluminación es muy importante al momento de realizar procesos que funcionen por medio de visión artificial y para evitar errores al momento de la programación, se requiere implementar un sistema de iluminación adecuado para su funcionamiento. A continuación, se detalla algunos tipos de iluminación utilizadas en visión artificial.

Retroiluminación Difusa: Este tipo de iluminación es la adecuada para visión artificial, cuando solamente se necesita el contorno de una pieza para su reconocimiento o medida, y para ello es necesario apoyar la pieza en una superficie transparente. Este sistema consiste en iluminar debajo del objeto a analizar y la cámara en la parte superior, proporcionando imágenes de alto contraste en la pieza y el fondo. Sus principales aplicaciones son utilizadas en inspección dimensional para calcular el tamaño del elemento y medir el nivel de porosidad de algunas sustancias [11].

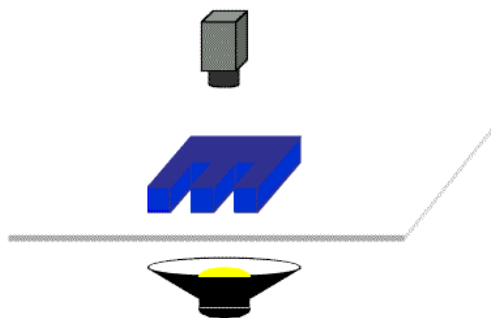


Figura 1.7. Retroiluminación difusa [11]

Iluminación frontal: Este tipo de iluminación en la mayoría de las aplicaciones es la más utilizada y se basa en iluminar frontalmente el elemento, pero en visión artificial presenta problemas al alcanzar un buen contraste entre el elemento y el fondo, debido al surgimiento de sombras y brillos que interfieren en las características del objeto. Sin embargo, este tipo de iluminación se usa en objetos poco reflectores para evitar brillos molestos [11].

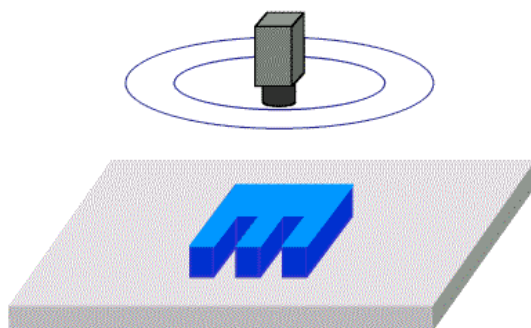


Figura 1.8. Iluminación frontal [11]

Luz Direccional: Es un tipo de Iluminación enfocada en una dirección específica en el espacio para resaltar una propiedad específica del objeto. Su mejor cualidad es crear sombras sobre el elemento lo que ayuda a incrementar el contraste de partes tridimensionales y adquirir información 3D [11].

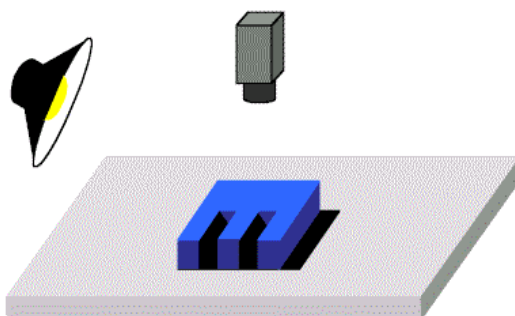


Figura 1.9. Luz direccional [11]

Luz Estructurada: Se basa en la proyección sobre un objeto de patrones de luz modulados y por medio de la luz reflejada se puede adquirir información sobre la estructura del área del elemento, la cual se reconstruye por medio de triangulación. La iluminación es especial ya que deben ser capaces de emitir luz estructurada y mayormente son láseres. Sus aplicaciones son para reconocimiento de la forma del objeto y reconstrucción 3D [11].

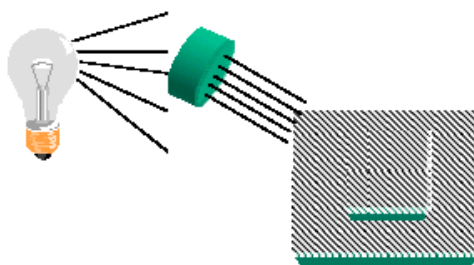


Figura 1.10. Luz estructurada [11]

Proceso de la Visión Artificial

El proceso de visión artificial es intervenido por una serie de pasos incluyendo métodos de software y hardware que empieza desde una imagen en un espacio específico. Ya con la superficie donde se encuentra el objeto a analizar, el primer paso es la adquisición de la imagen digital que en este caso es a través de una cámara web, misma que es el dispositivo con capacidad de digitalizar la imagen al computador.

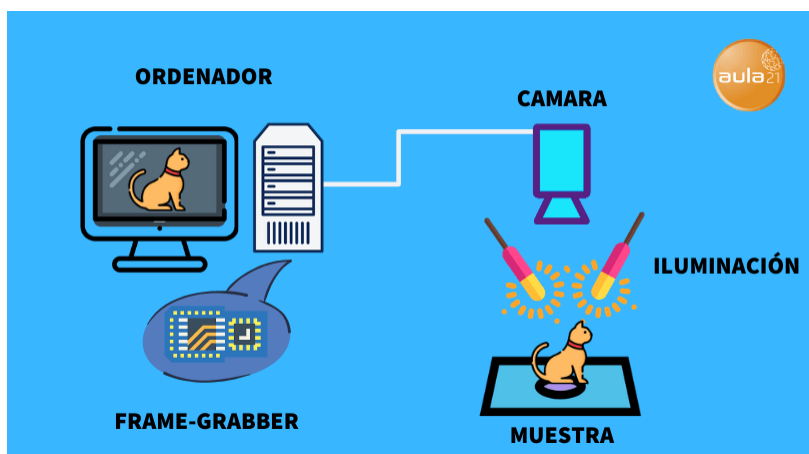


Figura 1.11. Proceso de Visión Artificial [12]

Al tener la imagen digitalizada, el paso que sigue es el filtrado o transformación de esta, en la cual se eliminan los rasgos ajenos y se mejora la imagen. A continuación, sigue la segmentación de la imagen para aislar los elementos ajenos a la misma, la cual se convertirá en datos que interprete el procesador. Luego se procede al cálculo de las características de la imagen, descomponiéndola en sus principales características en el proceso de segmentación, para extraer los rasgos específicos se usa el método de parametrización. Finalmente se procede con el reconocimiento y la interpretación de la imagen captada, donde la información recibida de la cámara se procesa y compara con la información previamente ingresada [13].



Figura 1.12. Proceso de un sistema de detección de monedas por visión artificial [14]

2 METODOLOGÍA

Para el desarrollo del presente proyecto se llevó a cabo la siguiente metodología: como primer paso, en la fase teórica, se estudió los conceptos fundamentales y los algoritmos de programación que se utilizan para visión artificial usando el lenguaje de programación “Python”. Se estudió la librería para visión artificial desarrollada por Intel denominada “OpenCV”. El segundo paso, en la fase de diseño, se determina los elementos necesarios del sistema y se desarrolla el algoritmo de programación para el mismo. Para el tercer paso, en la fase de simulación, se realizan las pruebas de funcionamiento de la cámara, así como del algoritmo desarrollado. Finalmente, como último paso, en la fase de análisis de resultados, se verifica el funcionamiento del sistema de detección de monedas estadounidenses y se realizan varias pruebas para corroborar el respectivo funcionamiento del sistema de visión artificial.

2.1 Requerimientos mínimos del sistema

Para el desarrollo del sistema de detección de monedas estadounidenses se determinó los siguientes elementos para su funcionamiento:

Cámara web

Para el funcionamiento del sistema, en este caso, se requiere de una cámara web básica para la adquisición de imágenes y detección de monedas estadounidenses. Para ello, se utilizará una cámara que ya se disponía anteriormente y a continuación, se presentan las características y especificaciones técnicas en la Tabla 2.1.



Figura 2.1. Cámara web

Tabla 2.1. Características y especificaciones de la cámara [15]

Características	
Fabricante	BRobotix
Modelo	651312
Color de Producto	Negro
Interfaz	USB 2.0
Micrófono Integrado	Sí
Especificaciones Técnicas	
Máxima resolución de video	1932 x 1088 Píxeles
Formato de video soportado	360p,720p,1080p
Ángulo de visión, horizontal	80°
Velocidad de foto	30 fps
Píxeles	2.0 Mega Píxeles
Longitud del cable	1,4 metros

Software de programación Pycharm

Es un IDE, es decir un editor de código, pero no solamente es eso ya que tiene un depurador, tiene un intérprete, así como otras herramientas que facilitarán la programación de algoritmos que se creen [16].

En este caso se usará el IDE “Pycharm” para desarrollar el código del sistema a implementar utilizando lenguaje de programación “Python”. En la Figura 2.2 se observa el IDE con el que se trabajará para realizar el programa para el sistema de reconocimiento de monedas estadounidenses.

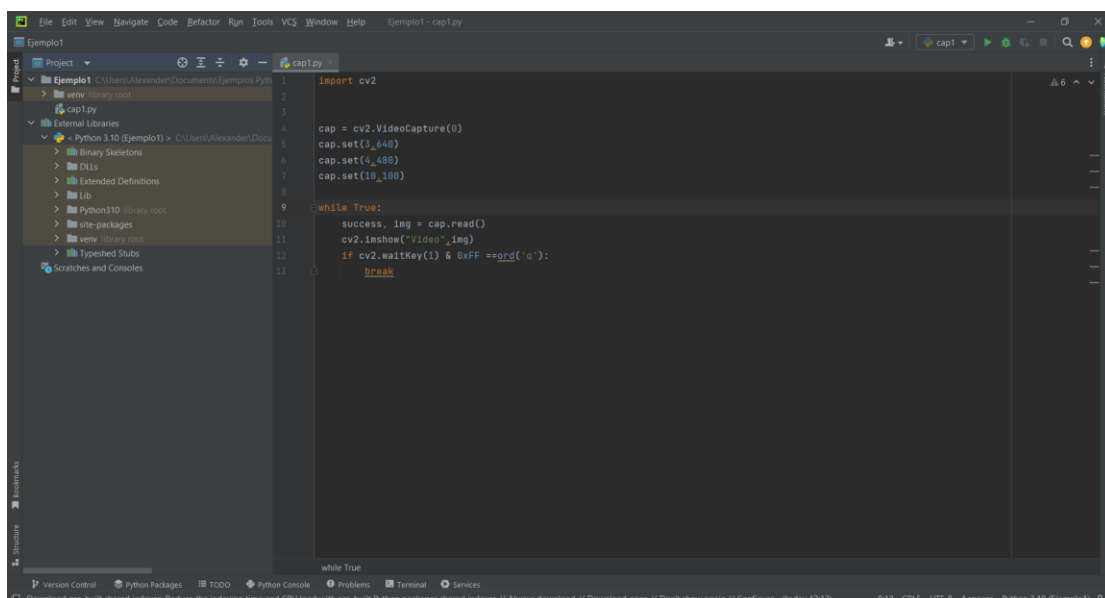


Figura 2.2. IDE "Pycharm"

Espacio de trabajo

Para determinar el espacio de trabajo se debe tomar en cuenta como se llevará a cabo el sistema de detección de monedas estadounidenses. En este caso, se elaboró una caja rectangular con tablas y espumas flex (es importante mencionar que estos materiales se disponían en casa). Las cuáles fueron pintadas de color negro, y dentro de la caja se iluminó con focos led utilizando iluminación indirecta para evitar brillos indeseados al momento de detectar las monedas estadounidenses ya que la zona de referencia será una hoja de papel bond tamaño A6 donde se procederá a colocar las monedas. En la Figura 2.3 se puede observar el espacio de trabajo donde se detectará las monedas estadounidenses.

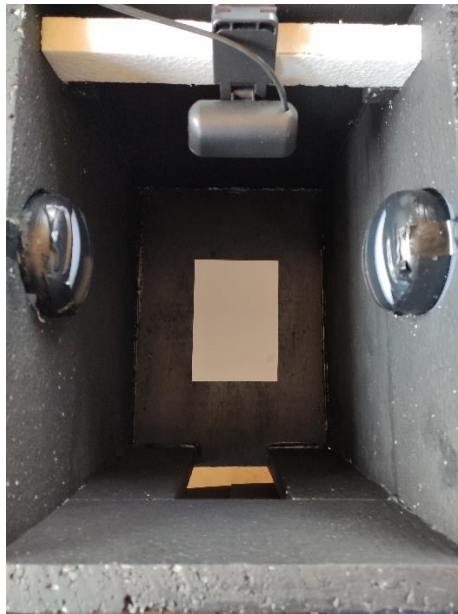


Figura 2.3. Espacio de Trabajo

2.2 Pruebas de cámara

Antes de empezar con el diseño del algoritmo, se procederá a revisar el funcionamiento de la cámara y para ello se usará un código, desarrollado en Pycharm, para encender la cámara.

En la Figura 2.4 se puede observar el código desarrollado en Pycharm para realizar las pruebas de funcionamiento de la cámara web. El objetivo de este código es encender y apagar la cámara, así como configurar sus principales características.

```

1  import cv2
2
3
4  cap = cv2.VideoCapture(0)
5  cap.set(3,640)
6  cap.set(4,480)
7  cap.set(10,100)
8
9  while True:
10     success, img = cap.read()
11     cv2.imshow("Video",img)
12     if cv2.waitKey(1) & 0xFF ==ord('q'):
13         break

```

Figura 2.4. Código para pruebas de funcionamiento de la cámara web

Una vez compilado y ejecutado el código se puede observar en la Figura 2.5 la prueba de funcionamiento de la cámara web utilizada.

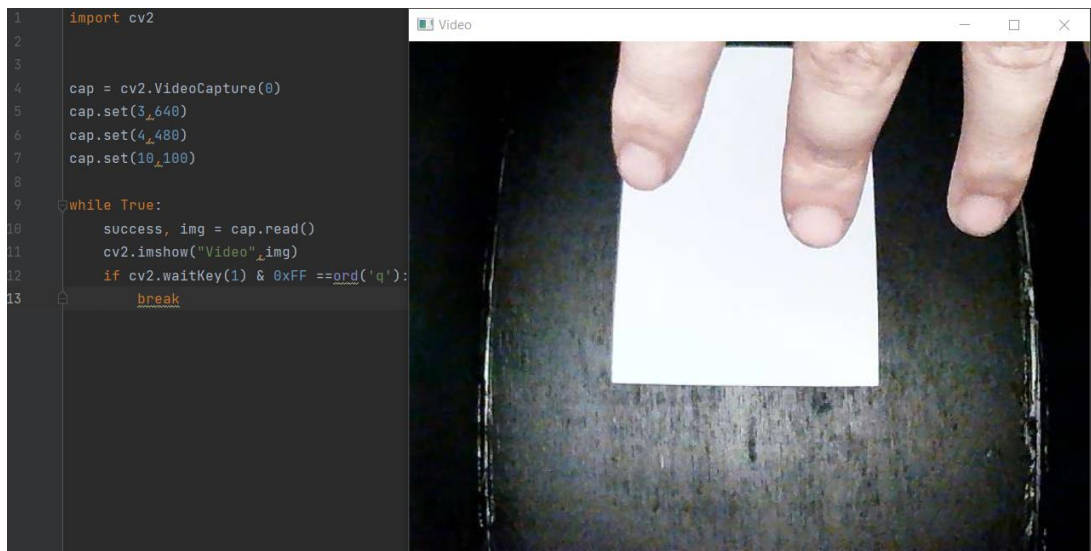


Figura 2.5. Funcionamiento de la cámara web

2.3 Diseño del Algoritmo

Desarrollo del algoritmo

El primer paso para realizar el algoritmo es importar las librerías a utilizar, como se mencionó en el capítulo anterior, las librerías a usar para la implementación del sistema de reconocimiento de monedas estadounidenses son OpenCV y Numpy. Se detalla las líneas de código realizado a continuación en la Figura 2.6.

```

1 import cv2
2 import numpy as np

```

Figura 2.6. Importar Librerías OpenCV y Numpy

Luego de proceder con la importación de las librerías que se trabajará, utilizamos la librería Numpy para crear matrices que puedan almacenar una cantidad de datos considerable, para que el algoritmo trabaje rápidamente al momento de guardar la información. Esto se observa en la Figura 2.7.

```

4 def ordenarPuntos (puntos):
5     n_puntos = np.concatenate([puntos[0], puntos[1], puntos[2], puntos[3]]).tolist()
6     y_order = sorted(n_puntos, key=lambda n_puntos: n_puntos[1])
7     x1_order = y_order[:2]
8     x1_order = sorted(x1_order, key=lambda x1_order: x1_order[0])
9     x2_order = y_order[2:4]
10    x2_order = sorted(x2_order, key=lambda x2_order: x2_order[0])
11
12    return [x1_order[0], x1_order[1], x2_order[0], x2_order[1]]

```

Figura 2.7. Uso de la Librería Numpy

Después empezamos a definir la imagen con la que se trabajará ya que al inicio se debe alinear la imagen para que el programa pueda reconocer con facilidad a las monedas dependiendo de su área. Se observa las líneas de código realizada en la Figura 2.8 a continuación:

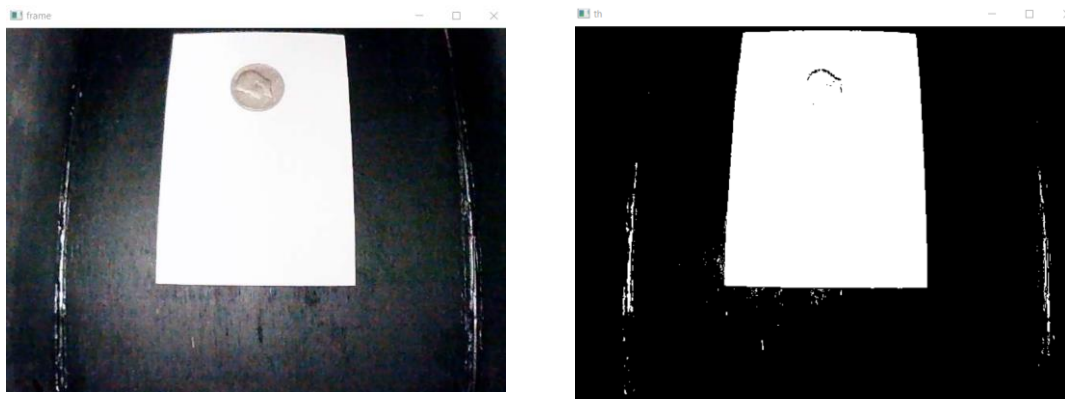
```

14 def roi(image, ancho, alto):
15     imagen_alineada = None
16     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
17     _th = cv2.threshold(gray, 150, 255, cv2.THRESH_BINARY)
18     cv2.imshow('th', _th)
19     cnts = cv2.findContours(_th, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[0]
20     cnts = sorted(cnts, key=cv2.contourArea, reverse=True)[:1]
21
22     for c in cnts:
23         epsilon = 0.01*cv2.arcLength(c, True)
24         approx = cv2.approxPolyDP(c, epsilon, True)
25
26         if len(approx) == 4:
27             puntos = ordenarPuntos(approx)
28             pts1 = np.float32(puntos)
29             pts2 = np.float32([[0,0],[ancho,0],[0,alto],[ancho,alto]])
30             M = cv2.getPerspectiveTransform(pts1, pts2)
31             imagen_alineada = cv2.warpPerspective(image, M, (ancho, alto))
32
33     return imagen_alineada

```

Figura 2.8. Proceso para alinear la imagen

Posteriormente, se procede a transformar la imagen original a escala de grises para después transformarlo a datos binarios que pueda leer el programa realizado.



a) Imagen original del sistema

b) Imagen a escala de grises

Figura 2.9. Proceso para transformar imágenes

Luego de realizado este procedimiento, se procederá a ordenar los puntos de la imagen, en este caso basada en una hoja tamaño A6 para que al momento de la imagen final se observe la imagen alineada.



a) Imagen original

b) Imagen Alineada

Figura 2.10. Comparación de la imagen sin alinear vs imagen alineada

En la Figura 2.11, se detalla parte del código que procederá a remarcar el contorno de las monedas estadounidenses para luego determinar el área de cada una, en base a una imagen en datos binarios, como se observa en la Figura 2.12.

```

35  cap = cv2.VideoCapture(0)
36
37  while True:
38      ret, frame = cap.read()
39      if ret == False: break
40
41      imagen_A6 = roi(frame, ancho=480, alto=640)
42      if imagen_A6 is not None:
43          puntos = []
44          imagen_gris = cv2.cvtColor(imagen_A6, cv2.COLOR_BGR2GRAY)
45          blur = cv2.GaussianBlur(imagen_gris, (5, 5), 1)
46          th_2 = cv2.threshold(blur, 0, 255, cv2.THRESH_OTSU+cv2.THRESH_BINARY_INV)
47          cv2.imshow('th_monedas', th_2)
48          cnts_2 = cv2.findContours(th_2, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[0]
49          cv2.drawContours(imagen_A6, cnts_2, -1, (255, 0, 0), 2)

```

Figura 2.11. Líneas de programación para remarcar el contorno de la moneda

En la Figura 2.12 se puede observar una imagen en blanco y negro en la cual se visualiza el área de la moneda en color blanco y el fondo en color negro. Es decir, que el fondo en donde se encuentra la moneda como es de color blanco se transforma mediante programación en color negro y al momento de colocar las monedas ya que son diferentes al color que se usó de fondo, se logrará observar su área de otro color en este caso de color blanco.

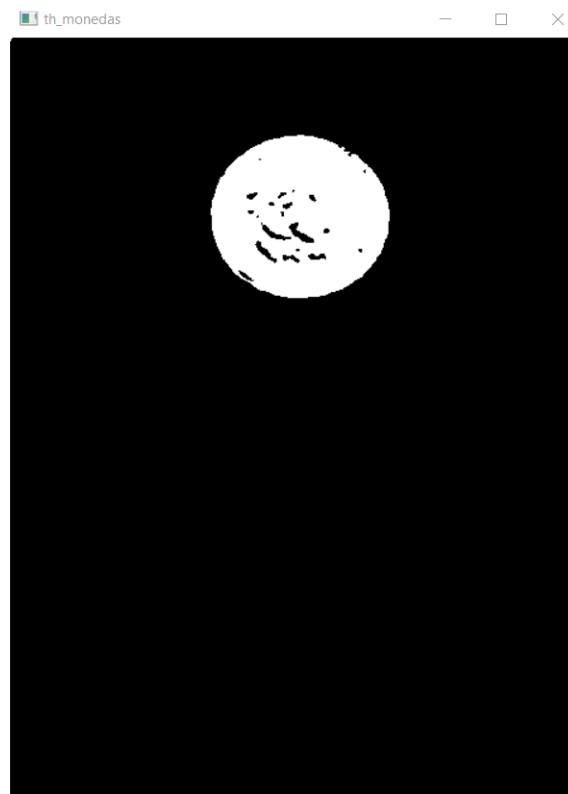


Figura 2.12. Imagen en Datos Binarios

En la Tabla 2.2 se detalla las condiciones mínimas y máximas de áreas de cada una de las monedas en base a la cuáles se realizará su detección. Así mismo, se indica el nombre que se visualizará al momento del reconocimiento.

Tabla 2.2. Rango de detección de áreas de las monedas estadounidenses

Monedas	Áreas (Píxeles Cuadrados)	
	Valor Máximo	Valor mínimo
50 centavos	16500	14650
Moneda Desconocida	14600	11925
1 dólar	11900	11175
Moneda Desconocida	11150	9850
25 centavos	9800	9400
Moneda Desconocida	9350	7625
5 centavos	7600	7000
Moneda Desconocida	6950	6050
1 centavo	6000	5600
Moneda Desconocida	5575	5425
10 centavos	5400	4850
Moneda Desconocida	4800	4000

Por ejemplo, para una moneda de 50 centavos, si el valor del área se encuentra entre 16000 y 14650 píxeles cuadrados en la detección marcará '50 ctvs', mientras que si el rango está entre 14600 y 11925 píxeles cuadrados se podrá visualizar 'Moneda Desconocida'

Luego, se procederá a colocar los detalles necesarios para que se detecte las monedas estadounidenses en la imagen final y así visualizar los contornos de las monedas con su respectivo valor, mismo que se mostrará en la parte inferior de la moneda en la imagen final. Esto se puede observar en la Figura 2.13.

```
if area < 16500 and area > 14650:
    font = cv2.FONT_HERSHEY_SIMPLEX
    cv2.putText(imagen_A6, '50 ctvs', (195, 290), font, 0.75, (0, 255, 0), 2)
```

Figura 2.13. Línea de código para visualización del valor de la moneda

En la Figura 2.14 se puede observar las líneas de código para colocar texto en la imagen final, y que su visualización sea más amigable para el usuario al momento de su funcionamiento.

```

line1 = cv2.line(imagen_A6, (240, 125), (240, 175), (0, 255, 255), 1)
line2 = cv2.line(imagen_A6, (215, 150), (265, 150), (0, 255, 255), 1)
line3 = cv2.line(imagen_A6, (0, 320), (480, 320), (0, 0, 200), 5)
fuente = cv2.FONT_HERSHEY_SIMPLEX
fuente2 = cv2.FONT_HERSHEY_PLAIN
fuente3 = cv2.FONT_HERSHEY_TRIPLEX
text1 = cv2.putText(imagen_A6, 'SISTEMA DE DETECCION DE', (25, 380), fuente, 1, (0, 0, 0), 1, cv2.LINE_AA)
text2 = cv2.putText(imagen_A6, 'MONEDAS ESTADOUNIDENSES', (13, 420), fuente, 1, (0, 0, 0), 1, cv2.LINE_AA)
text3 = cv2.putText(imagen_A6, 'Alexander David Andrango Ayo', (2, 600), fuente, 1, (0, 0, 0), 1, cv2.LINE_AA)
text4 = cv2.putText(imagen_A6, 'Colocar la moneda en la cruz (+)', (5, 15), fuente2, 1, (0, 0, 0), 1, cv2.LINE_AA)
text5 = cv2.putText(imagen_A6, 'TECNOLOGIA SUPERIOR', (45, 500), fuente3, 1, (0, 0, 0), 1, cv2.LINE_AA)
text6 = cv2.putText(imagen_A6, 'EN ELECTROMECHANICA', (52, 530), fuente3, 1, (0, 0, 0), 1, cv2.LINE_AA)

```

Figura 2.14. Línea de código para colocar texto en la imagen final

En la Figura 2.15 se puede observar la imagen de presentación final del algoritmo implementado en la que se podrá visualizar el sistema de detección de monedas estadounidenses, el valor de la moneda detectado se encuentra de color verde.

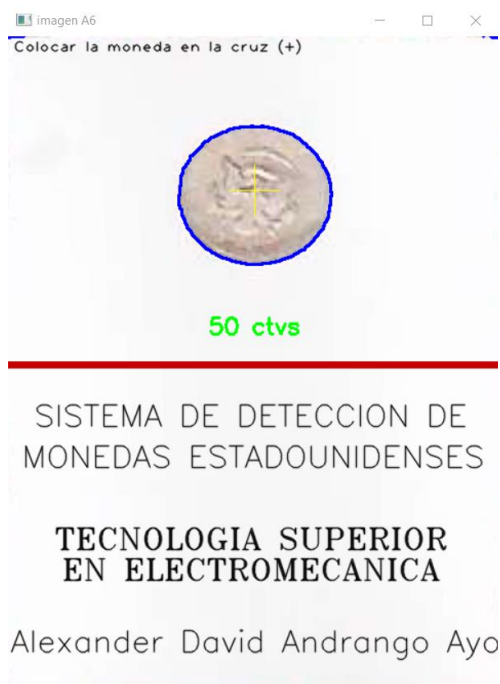


Figura 2.15. Imagen de reconocimiento de monedas estadounidenses

Finalmente, se implementó en el programa una forma de pausar el sistema para mejorar la detección de las monedas estadounidenses y sea más amigable para el usuario que lo utilice y que no exista falsas detecciones de las monedas.

Este sistema se basa en que al presionar la tecla espacio se pausará el programa y se podrá visualizar de una mejor manera la detección y para que vuelva a seguir corriendo se presionará cualquier otra tecla, además que con la tecla 'q' se finalizará el programa. A continuación, se presenta el código implementado en la Figura 2.16.

```
k = cv2.waitKey(1)
if k == 32:
    cv2.waitKey()
elif k == ord('q'):
    break

cap.release()
cv2.destroyAllWindows()
```

Figura 2.16. Línea de código para pausar y finalizar el programa de sistema de detección de monedas estadounidenses

3 RESULTADOS

Para analizar el funcionamiento del sistema de detección de monedas estadounidenses, se procederá a realizar varias pruebas, mismas que serán analizadas y así comprobar la detección de cada una de las monedas estadounidenses. Así mismo, se realizarán pruebas con monedas de otros países

3.1 Moneda de 1 dólar

Para verificar el reconocimiento de la moneda de 1 dólar, en la Tabla 3.1 se detallan los resultados obtenidos al realizar 10 pruebas de detección diferentes.

Tabla 3.1. Pruebas de detección de la moneda estadounidense de 1 dólar

Moneda de 1 dólar	
Prueba 1	Sí Detecta
Prueba 2	Sí Detecta
Prueba 3	Sí Detecta
Prueba 4	Sí Detecta
Prueba 5	Sí Detecta
Prueba 6	Sí Detecta
Prueba 7	Sí Detecta
Prueba 8	Sí Detecta
Prueba 9	Sí Detecta
Prueba 10	Sí Detecta

Por otra parte, en la Figura 3.1, se puede observar la imagen de visualización del algoritmo de detección para la moneda de 1 dólar.



Figura 3.1. Detección de moneda de 1 dólar

3.2 Moneda de 50 centavos

Para verificar el reconocimiento de la moneda de 50 centavos, en la Tabla 3.2 se detallan los resultados obtenidos al realizar 10 pruebas de detección diferentes.

Tabla 3.2. Pruebas de detección de la moneda estadounidense de 50 centavos

Moneda de 50 centavos	
Prueba 1	Sí Detecta
Prueba 2	Sí Detecta
Prueba 3	Sí Detecta
Prueba 4	Sí Detecta
Prueba 5	Sí Detecta
Prueba 6	Sí Detecta
Prueba 7	Sí Detecta
Prueba 8	Sí Detecta
Prueba 9	Sí Detecta
Prueba 10	Sí Detecta

Por otra parte, en la Figura 3.2, se puede observar la imagen de visualización del algoritmo de detección para la moneda de 50 centavos.

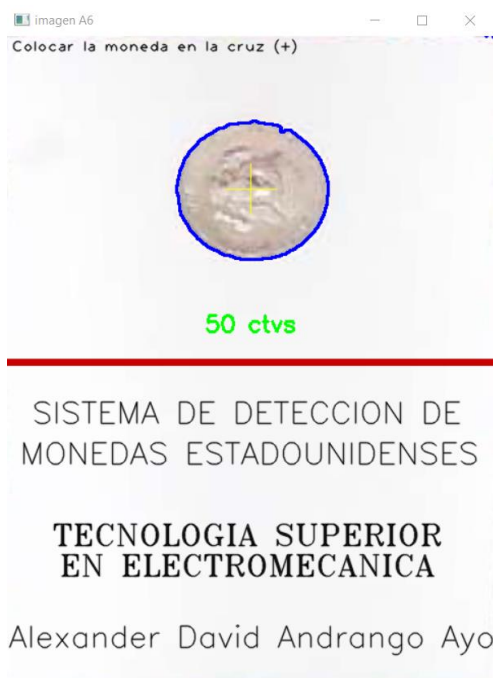


Figura 3.2. Detección de moneda de 50 centavos

3.3 Moneda de 25 centavos

Para verificar el reconocimiento de la moneda de 25 centavos, en la Tabla 3.3 se detallan los resultados obtenidos al realizar 10 pruebas de detección diferentes.

Tabla 3.3. Pruebas de detección de la moneda estadounidense de 25 centavos

Moneda de 25 centavos	
Prueba 1	Sí Detecta
Prueba 2	Sí Detecta
Prueba 3	Sí Detecta
Prueba 4	Sí Detecta
Prueba 5	Sí Detecta
Prueba 6	Sí Detecta
Prueba 7	Sí Detecta
Prueba 8	Sí Detecta
Prueba 9	Sí Detecta
Prueba 10	Sí Detecta

Por otra parte, en la Figura 3.3, se puede observar la imagen de visualización del algoritmo de detección para la moneda de 25 centavos.

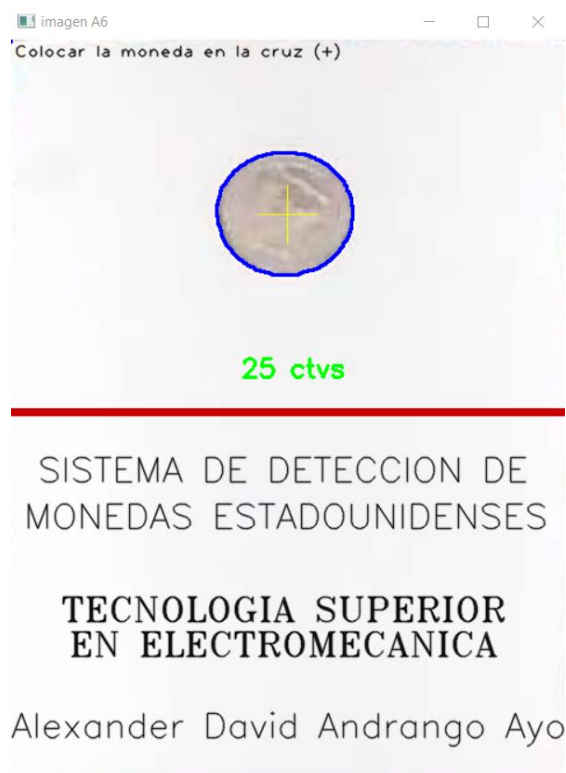


Figura 3.3. Detección de moneda de 25 centavos

3.4 Moneda de 10 centavos

Para verificar el reconocimiento de la moneda de 10 centavos, en la Tabla 3.4 se detallan los resultados obtenidos al realizar 10 pruebas de detección diferentes.

Tabla 3.4. Pruebas de detección de la moneda estadounidense de 10 centavos

Moneda de 10 centavos	
Prueba 1	Sí Detecta
Prueba 2	Sí Detecta
Prueba 3	Sí Detecta
Prueba 4	Sí Detecta
Prueba 5	Sí Detecta
Prueba 6	Sí Detecta
Prueba 7	Sí Detecta
Prueba 8	Sí Detecta
Prueba 9	Sí Detecta
Prueba 10	Sí Detecta

Por otra parte, en la Figura 3.4, se puede observar la imagen de visualización del algoritmo de detección para la moneda de 10 centavos.

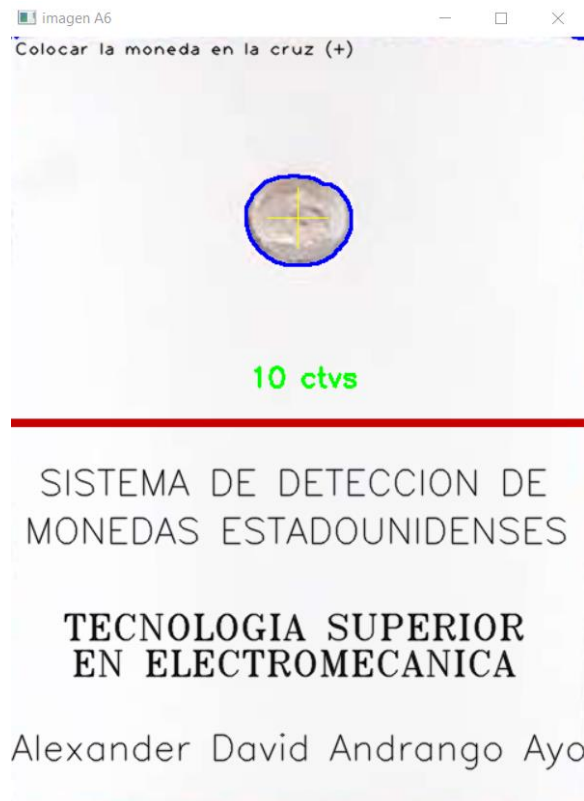


Figura 3.4. Detección de moneda de 10 centavos

3.5 Moneda de 5 centavos

Para verificar el reconocimiento de la moneda de 5 centavos, en la Tabla 3.5 se detallan los resultados obtenidos al realizar 10 pruebas de detección diferentes.

Tabla 3.5. Pruebas de detección de la moneda estadounidense de 5 centavos

Moneda de 5 centavos	
Prueba 1	Sí Detecta
Prueba 2	Sí Detecta
Prueba 3	Sí Detecta
Prueba 4	Sí Detecta
Prueba 5	Sí Detecta
Prueba 6	Sí Detecta
Prueba 7	Sí Detecta
Prueba 8	Sí Detecta
Prueba 9	Sí Detecta
Prueba 10	Sí Detecta

Por otra parte, en la Figura 3.5, se puede observar la imagen de visualización del algoritmo de detección para la moneda de 5 centavos.



Figura 3.5. Detección de moneda de 5 centavos

3.6 Moneda de 1 centavo

Para verificar el reconocimiento de la moneda de 1 centavo, en la Tabla 3.6 se detallan los resultados obtenidos al realizar 10 pruebas de detección diferentes.

Tabla 3.6. Pruebas de detección de la moneda estadounidense de 1 centavo

Moneda de 1 centavo	
Prueba 1	Sí Detecta
Prueba 2	Sí Detecta
Prueba 3	Sí Detecta
Prueba 4	Sí Detecta
Prueba 5	Sí Detecta
Prueba 6	Sí Detecta
Prueba 7	Sí Detecta
Prueba 8	Sí Detecta
Prueba 9	Sí Detecta
Prueba 10	Sí Detecta

Por otra parte, en la Figura 3.6, se puede observar la imagen de visualización del algoritmo de detección para la moneda de 1 centavo.



Figura 3.6. Detección de moneda de 1 centavo

3.7 Monedas de otros países

Además, en el sistema de detección de monedas estadounidenses se implementó líneas de código para que, al momento de detectar otras monedas de diferentes países, las cuales tienen otras dimensiones, el sistema las detectará como 'Moneda desconocida'. En la Tabla 3.7 se observan los resultados obtenidos con cada una de las monedas trabajadas.

Tabla 3.7. Pruebas de detección de otras monedas

Monedas de otros países				
Moneda	País/Región	Prueba 1	Prueba 2	Prueba 3
1 real	Brasil	✓	✓	✓
50 centavos		✓	✓	✓
25 centavos		✓	✓	✓
10 centavos		✓	✓	✓
1 nuevo sol	Perú	✓	✓	✓
10 céntimos		✓	✓	✓
20 céntimos		✓	✓	✓
50 céntimos		✓	✓	✓
50 centavos	Bolivia	✓	✓	✓
1 peso	Argentina	✓	✓	✓
1 peso	Chile	✓	✓	✓
2 pesos	México	✓	✓	✓
500 pesos	Colombia	✓	✓	✓
20 euro cent	Europa	✓	✓	✓
1 euro cent		✓	✓	✓

En la Figura 3.7 se puede observar los resultados que muestra el software de detección para dos monedas extranjeras, mismas que son de 50 centavos de Bolivia y 20 centavos de euro, como resultado se observa el mensaje 'Moneda Desconocida'.



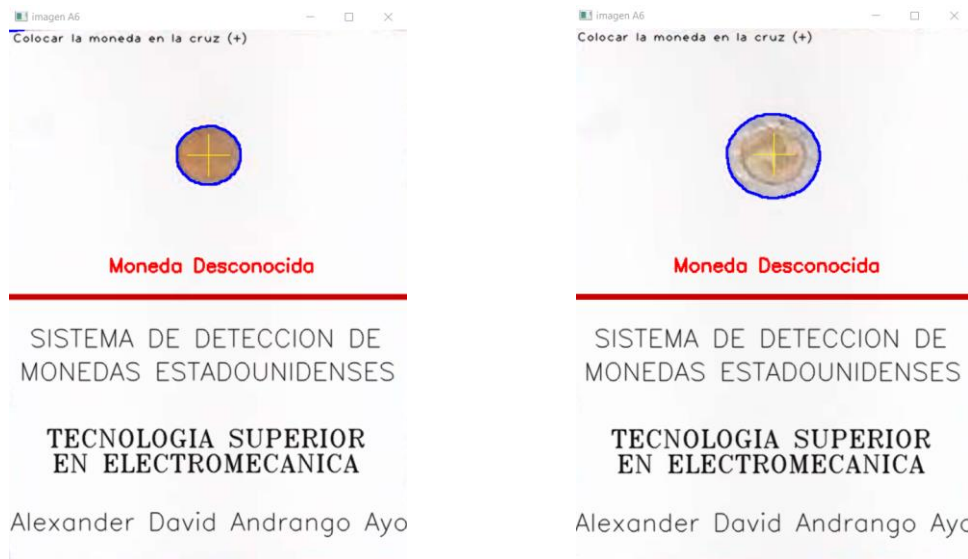
a) Moneda 50 centavos de Bolivia



b) Moneda 20 centavos de euro

Figura 3.7. Detección de monedas de otros países

Adicionalmente, en la Figura 3.8 se muestra la visualización de dos pruebas adicionales, en este caso de la moneda de 2 centavos de euro y la moneda de 2 pesos mexicanos. Como se puede observar, en el software se visualiza el mensaje 'Moneda Desconocida' en la parte inferior de la moneda detectada.



a) Moneda de 2 centavos de euro

b) Moneda de 2 pesos de México

Figura 3.8. Detección de monedas de otros países

En el Anexo V se adjunta el enlace del video, del funcionamiento del sistema de detección de monedas estadounidenses por visión artificial.

4 CONCLUSIONES

La visión artificial hoy en día es muy importante, ya que facilita la automatización de procesos industriales y que estos sean más rápidos. En este trabajo se implementó un sistema de detección de monedas estadounidenses para entender y comprender el principio básico de funcionamiento de la detección de patrones simples en una aplicación de un sistema basado en visión artificial.

Una de las etapas primordiales para la implementación de un sistema de visión artificial es determinar los requerimientos mínimos, para este proyecto se contó con una cámara web, el espacio de trabajo y el software IDE 'Pycharm' en el cual se desarrolló el algoritmo de funcionamiento. Es importante que todos estos elementos sean acordes al sistema que se va a implementar para su correcto funcionamiento.

Al momento de desarrollar el algoritmo, es importante conocer las etapas que se deben cumplir al momento de la detección de monedas estadounidenses. Además, de que el cuadro de visualización de la imagen final del sistema sea el adecuado, y que este sea amigable al usuario que lo va a utilizar para observar el funcionamiento.

Para las pruebas de funcionamiento se realizaron varias, con el objetivo de comprobar que el sistema de detección de monedas estadounidenses trabaje correctamente. Para ello se realizaron 10 pruebas de funcionamiento con cada una de las monedas estadounidenses obteniendo como resultado una detección satisfactoria para todos los casos de prueba.

Además, se realizaron varias pruebas con monedas de otros países, ya que dependiendo de su área el algoritmo visualizará un mensaje como 'Moneda Desconocida' en la imagen final, obteniendo un resultado satisfactorio con cada una de las monedas de otros países con las que se comprobaron.

5 RECOMENDACIONES

Es importante que dependiendo del sistema de visión artificial con el cual se va a trabajar, se tome en cuenta un sistema de iluminación adecuado, en vista de que la obtención del tipo de imágenes que uno requiere va a depender del tipo de iluminación seleccionado. Como se mostró en este trabajo existen ya estudios e investigaciones sobre los tipos de iluminación acorde a los patrones a detectar, mismos que pueden ayudar a mejorar el proyecto que se va a realizar.

Para el proyecto de detección de monedas estadounidenses, es importante acondicionar el sistema dependiendo del patrón que se va a utilizar para la respectiva detección. Para este proyecto se trabajó con el área de las monedas, en este caso, se debe tener mucho cuidado al ingresar los rangos de medidas de área para que el algoritmo detecte a la moneda correctamente.

Se debe tomar en cuenta que todo software debe ser intuitivo con el usuario, debido a que éste debe poder utilizar el software de manera sencilla y sea fácil de comprender su funcionamiento. En este caso, se colocaron mensajes en la imagen final del sistema para que el usuario identifique claramente que la moneda fue identificada.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] “CONTAVAL.” <https://www.contaval.es/que-es-la-vision-artificial-y-para-que-sirve/> (accessed Jun. 05, 2022).
- [2] “¿Qué es la Visión artificial?” <https://www.intel.es/content/www/es/es/manufacturing/what-is-machine-vision.html> (accessed Jun. 05, 2022).
- [3] “Sistemas de Visión Artificial Industrial: 3 Tipos y Aplicaciones ✓ 【2022】 .” <https://www.edsrobotics.com/blog/sistemas-de-vision-artificial-tipos-aplicaciones/> (accessed Jun. 12, 2022).
- [4] “Python: qué es, para qué sirve y cómo se programa | Informática Industrial.” <https://www.cursosaula21.com/que-es-python/> (accessed Jun. 05, 2022).
- [5] “¿Qué es Python? | Blog Becas Santander.” <https://www.becas-santander.com/es/blog/python-que-es.html> (accessed Jun. 05, 2022).
- [6] “OpenCV.” <http://acodigo.blogspot.com/p/tutorial-opencv.html> (accessed Jun. 05, 2022).
- [7] “Getting Started on Object Detection with OpenCV | by Collins Hillary | Analytics Vidhya | Medium.” <https://medium.com/analytics-vidhya/getting-started-on-object-detection-with-opencv-5962a75876a6> (accessed Jun. 12, 2022).
- [8] “La librería Numpy | Aprende con Alf.” <https://aprendeconalf.es/docencia/python/manual/numpy/> (accessed Jun. 12, 2022).
- [9] “Dólar americano - Global Exchange Ecuador.” <https://www.globalexchange.ec/monedas-del-mundo/dolar-americano> (accessed Jun. 05, 2022).
- [10] “→ Camara Web↓ 【Que es, para que sirve y cual es su funcion】 .” https://partesdelacomputadora.org/externas/camara-web/#Lente_optico (accessed Jun. 05, 2022).
- [11] E. Sobrado, “SISTEMA DE VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO Y MANIPULACIÓN DE OBJETOS UTILIZANDO UN BRAZO ROBOT,” PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ, Lima, 2003. Accessed: Jun.

- 12, 2022. [Online]. Available:
<https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/68>
- [12] “Visión Artificial: todo lo que necesitas saber.”
<https://www.cursosaula21.com/que-es-la-vision-artificial/> (accessed Aug. 14, 2022).
- [13] Grupo de Investigación EDMANS, “Técnicas y algoritmos básicos de visión artificial,” *Universidad de La Rioja*. Servicio de Publicaciones, 2006. Accessed: Jun. 12, 2022. [Online]. Available:
<https://publicaciones.unirioja.es/catalogo/online/VisionArtificial.pdf>
- [14] L. Manobanda, “IMPLEMENTACIÓN DE UN PROTOTIPO DE MÁQUINA CLASIFICADORA Y CONTADORA DE MONEDAS BASADA EN VISIÓN ARTIFICIAL,” ESCUELA SUPERIOR POLITÉCNICA DE CHIMORAZO, Riobamba, Ecuador, 2019. Accessed: May 19, 2022. [Online]. Available:
<http://dspace.esPOCH.edu.ec/handle/123456789/13514>
- [15] “Cámara Web BRobotix 651312 2 MP USB Micrófono Negro.”
<https://intercompras.com/p/camara-web-brobotix-mp-usb-microfono-negro-212389> (accessed Jun. 12, 2022).
- [16] “PyCharm, un potente IDE para crear programas con Python | Linux Adictos.”
<https://www.linuxadictos.com/pycharm-un-potente-ide-para-crear-programas-con-python.html> (accessed Jun. 12, 2022).

7 ANEXOS

ANEXO I

Certificado de Turnitin



**ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"**

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 22 de agosto de 2022

De mi consideración:

Yo, ARACELY INÉS YANDÚN TORRES, en calidad de Directora del Trabajo de Integración Curricular titulado SISTEMA DE DETECCIÓN DE PATRONES SIMPLES DE MONEDAS asociado al proyecto IMPLEMENTACIÓN DE UN SISTEMA DE DETECCIÓN DE MONEDAS POR VISIÓN ARTIFICIAL elaborado por el estudiante ALEXANDER DAVID ANDRANGO AYO de la carrera en TECNOLOGÍA SUPERIOR EN ELECTROMECAÁNICA, certifico que he empleado la herramienta Turnitin para la revisión de originalidad de las secciones (carátula, resumen, abstract, capítulo 1, capítulo 2, capítulo 3, capítulo 4 y capítulo 5) del documento escrito producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 11%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,

**Ing. Aracely Yandún Torres, Mgs.
Docente
ESFOT**

TIC - Alexander Andrango Ayo

INFORME DE ORIGINALIDAD

11%

INDICE DE SIMILITUD

11%

FUENTES DE INTERNET

2%

PUBLICACIONES

2%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	dspace.ucuenca.edu.ec Fuente de Internet	1%
2	Submitted to Escuela Politecnica Nacional Trabajo del estudiante	1%
3	es.scribd.com Fuente de Internet	1%
4	uvadoc.uva.es Fuente de Internet	1%
5	bibdigital.epn.edu.ec Fuente de Internet	1%
6	es.slideshare.net Fuente de Internet	1%
7	www.coursehero.com Fuente de Internet	<1%
8	Submitted to Pontificia Universidad Catolica del Peru Trabajo del estudiante	<1%
9	itla.edu.do Fuente de Internet	

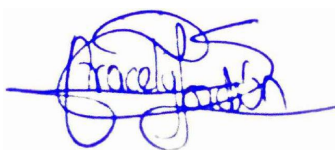
ANEXO II

Certificado de funcionamiento de Trabajo de Integración Curricular

DMQ, 22 de agosto de 2022

Yo, Aracely Inés Yandún Torres, docente a tiempo completo de la Escuela Politécnica Nacional y como directora de este trabajo de integración curricular, certifico que he constatado el correcto funcionamiento del entregable del proyecto de titulación **IMPLEMENTACIÓN DE UN SISTEMA DE DETECCIÓN DE MONEDAS POR VISIÓN ARTIFICIAL**, el cual fue desarrollado por el estudiante Alexander David Andrango Ayo.

El proyecto cumple con los requerimientos de diseño y parámetros necesarios para que los usuarios de la ESFOT puedan usarlo sin inconvenientes.



DIRECTOR

Ing. Aracely Inés Yandún Torres, Mgs.

Ladrón de Guevara E11-253 y Andalucía | Edificio N. 21 | Oficina 2 – Módulo 12

Correo: aracely.yandun@epn.edu.ec | **Ext:** 2711

ANEXO III

Líneas de Código Desarrollado del Sistema de Detección de monedas estadounidenses por visión artificial

```
import cv2
import numpy as np

def ordenarPuntos (puntos):
    n_puntos = np.concatenate([puntos[0], puntos[1], puntos[2], puntos[3]]).tolist()
    y_order = sorted(n_puntos, key=lambda n_puntos: n_puntos[1])
    x1_order = y_order[:2]
    x1_order = sorted(x1_order, key=lambda x1_order: x1_order[0])
    x2_order = y_order[2:4]
    x2_order = sorted(x2_order, key=lambda x2_order: x2_order[0])

    return [x1_order[0], x1_order[1], x2_order[0], x2_order[1]]

def roi(image, ancho, alto):
    imagen_alineada = None
    gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
    _,th = cv2.threshold(gray, 150, 255, cv2.THRESH_BINARY)
    cnts = cv2.findContours(th,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)[0]
    cnts = sorted(cnts, key=cv2.contourArea, reverse=True)[:1]

    for c in cnts:
        epsilon = 0.01*cv2.arcLength(c,True)
        approx = cv2.approxPolyDP(c,epsilon,True)

        if len(approx) == 4:
            puntos = ordenarPuntos(approx)
            pts1 = np.float32(puntos)
            pts2 = np.float32([[0,0],[ancho,0],[0,alto],[ancho,alto]])
            M = cv2.getPerspectiveTransform(pts1,pts2)
```

```

imagen_alineada = cv2.warpPerspective(image,M,(ancho,alto))

return imagen_alineada

cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    if ret == False:break

imagen_A6 = roi(frame, ancho=480, alto=640)
if imagen_A6 is not None:
    puntos = []
    imagen_gris = cv2.cvtColor(imagen_A6, cv2.COLOR_BGR2GRAY)
    blur = cv2.GaussianBlur(imagen_gris, (5, 5), 1)
    _, th_2 = cv2.threshold(blur, 0, 255, cv2.THRESH_OTSU+cv2.THRESH_BINARY_INV)
    cnts_2 = cv2.findContours(th_2, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[0]
    cv2.drawContours(imagen_A6,cnts_2, -1, (255, 0, 0), 2)

    for c_2 in cnts_2:

        area = cv2.contourArea(c_2)
        Momentos = cv2.moments(c_2)

        if (Momentos["m00"]==0):
            Momentos["m00"] = 1.0

        x = int(Momentos["m10"]/Momentos["m00"])
        y = int(Momentos["m01"] / Momentos["m00"])

        if area < 16500 and area > 14650:
            font = cv2.FONT_HERSHEY_SIMPLEX

```

```
cv2.putText(imagen_A6, '50 ctvs', (195, 290), font, 0.75, (0, 255, 0), 2)
```

```
elif area < 14600 and area > 11925:
```

```
font = cv2.FONT_HERSHEY_SIMPLEX
```

```
cv2.putText(imagen_A6, 'Moneda Desconocida', (120, 290), font, 0.75, (0, 0, 255), 2)
```

```
elif area < 11900 and area > 11175:
```

```
font = cv2.FONT_HERSHEY_SIMPLEX
```

```
cv2.putText(imagen_A6, '1$', (220, 290), font, 0.75, (0, 255, 0), 2)
```

```
elif area < 11150 and area > 9850:
```

```
font = cv2.FONT_HERSHEY_SIMPLEX
```

```
cv2.putText(imagen_A6, 'Moneda Desconocida', (120, 290), font, 0.75, (0, 0, 255), 2)
```

```
elif area < 9800 and area > 9400:
```

```
font = cv2.FONT_HERSHEY_SIMPLEX
```

```
cv2.putText(imagen_A6, '25 ctvs', (200, 290), font, 0.75, (0, 255, 0), 2)
```

```
elif area < 9350 and area > 7625:
```

```
font = cv2.FONT_HERSHEY_SIMPLEX
```

```
cv2.putText(imagen_A6, 'Moneda Desconocida', (120, 290), font, 0.75, (0, 0, 255), 2)
```

```
elif area < 7600 and area > 7000:
```

```
font = cv2.FONT_HERSHEY_SIMPLEX
```

```
cv2.putText(imagen_A6, '5 ctvs', (205, 290), font, 0.75, (0, 255, 0), 2)
```

```
elif area < 6950 and area > 6050:
```

```
font = cv2.FONT_HERSHEY_SIMPLEX
```

```
cv2.putText(imagen_A6, 'Moneda Desconocida', (120, 290), font, 0.75, (0, 0, 255), 2)
```

```
elif area < 6000 and area > 5600:
```

```
font = cv2.FONT_HERSHEY_SIMPLEX
```



```

cv2.putText(imagen_A6, '1 ctv', (210, 290), font, 0.75, (0, 255, 0), 2)

elif area < 5575 and area > 5425:
    font = cv2.FONT_HERSHEY_SIMPLEX
    cv2.putText(imagen_A6, 'Moneda Desconocida', (120, 290), font, 0.75, (0, 0, 255), 2)

elif area < 5400 and area > 4850:
    font = cv2.FONT_HERSHEY_SIMPLEX
    cv2.putText(imagen_A6, '10 ctvs', (200, 290), font, 0.75, (0, 255, 0), 2)

elif area < 4800 and area > 4000:
    font = cv2.FONT_HERSHEY_SIMPLEX
    cv2.putText(imagen_A6, 'Moneda Desconocida', (120, 290), font, 0.75, (0, 0, 255), 2)

line1 = cv2.line(imagen_A6, (240, 125), (240, 175), (0, 255, 255), 1)
line2 = cv2.line(imagen_A6, (215, 150), (265, 150), (0, 255, 255), 1)
line3 = cv2.line(imagen_A6, (0, 320), (480, 320), (0, 0, 200), 5)
fuente = cv2.FONT_HERSHEY_SIMPLEX
fuente2 = cv2.FONT_HERSHEY_PLAIN
fuente3 = cv2.FONT_HERSHEY_TRIPLEX

text1 = cv2.putText(imagen_A6, 'SISTEMA DE DETECCION DE', (25, 380), fuente, 1, (0, 0, 0), 1,
cv2.LINE_AA)

text2 = cv2.putText(imagen_A6, 'MONEDAS ESTADOUNIDENSES', (13, 420), fuente, 1, (0, 0, 0), 1,
cv2.LINE_AA)

text3 = cv2.putText(imagen_A6, 'Alexander David Andrango Ayo', (2, 600), fuente, 1, (0, 0, 0), 1,
cv2.LINE_AA)

text4 = cv2.putText(imagen_A6, 'Colocar la moneda en la cruz (+)', (5, 15), fuente2, 1, (0, 0, 0), 1,
cv2.LINE_AA)

text5 = cv2.putText(imagen_A6, 'TECNOLOGIA SUPERIOR', (45, 500), fuente3, 1, (0, 0, 0), 1,
cv2.LINE_AA)

text6 = cv2.putText(imagen_A6, 'EN ELECTROMECHANICA', (52, 530), fuente3, 1, (0, 0, 0), 1,
cv2.LINE_AA)

cv2.imshow('imagen A6', imagen_A6)

```

```
k = cv2.waitKey(1)
if k == 32:
    cv2.waitKey()
elif k == ord('q'):
    break

cap.release()
cv2.destroyAllWindows()
```

ANEXO IV

Datasheet de la Cámara Web Utilizada

CÁMARA WEB FULL HD 1080P CON MICRÓFONO

Modelo 651312



Descripción:

- ▶ Cámara web diseñada y optimizada específicamente para transmisión de video en sitios de juegos sociales y entretenimiento como SKYPE, ZOOM y YouTube.
- La cámara le permite disfrutar de una voz más clara para una comunicación sin problemas. El micrófono integrado hace que su llamada de voz sea más clara. ¡Disfruta de su de alta definición Full HD 1080P!

Características:

- ▶ Cámara web diseñada y optimizada para transmisión de video.
- ▶ La función de clip permite colocarla fácilmente sobre el monitor de la computadora, TV o laptop. También cuenta con sistema de rosca para colocarse en trípode, o puede colocarse directamente sobre el escritorio.
- ▶ Ideal para realizar streaming profesional.
- ▶ Contenido: 1 Cámara Web, 1 manual de instrucciones.

Especificaciones:

- ▶ Cámara web de alta definición Full HD 1080P.
- ▶ Enfoque manual.
- ▶ Velocidad de foto: 25fps / 30fps
- ▶ Píxeles: 2.0 Mega Píxeles.
- ▶ Ángulo de visión: 80°
- ▶ Micrófono integrado
- ▶ Interfaz: USB V2.0
- ▶ Píxeles efectivos: 1932 (H) x 1088 (V)
- ▶ Resolución horizontal: 200 W.
- ▶ Relación señal a ruido: > 50 dB.
- ▶ UV: 640 x 360.
- ▶ Obturador electrónico: automático.
- ▶ Acuerdo: UVC / UVA.
- ▶ Salida de video: transmisión de video USB
- ▶ Voltaje de funcionamiento: CC 5 V, 160 mA ± 6 mA.
- ▶ Flujo de video: 1920 x 1080/1920 x 720/640 x 360
- ▶ MJPEG: 1920 x 1080/1920 x 720/640 x 360.
- ▶ Compresión automática del color.
- ▶ Rotación de 120 grados.
- ▶ Compatible con Vista / Win7 / Win8 / Win10 / Win11, Android y Mac OS.
- ▶ Plug and play.



ANEXO V

Enlace del Video de Funcionamiento del sistema

<https://youtu.be/MITkUgnFAEM>

The thumbnail features a blue background with white text. On the right, there is a screenshot of a software interface showing a coin being detected. The interface includes a red crosshair, a green '\$' symbol, and a red horizontal line. Below the screenshot, the text reads: 'SISTEMA DE DETECCION DE MONEDAS ESTADOUNIDENSES', 'TECNOLOGIA SUPERIOR EN ELECTROMECHANICA', and 'Alexander David Andrango Ayc'. The left side of the thumbnail contains the title 'IMPLEMENTACIÓN DE UN SISTEMA DE DETECCIÓN DE MONEDAS POR VISIÓN ARTIFICIAL', the subtitle 'TRABAJO DE INTEGRACIÓN CURRICULAR', and logos for 'ESCUELA POLITÉCNICA NACIONAL' and 'ESFOT ESCUELA POLITÉCNICA NACIONAL'.

Código QR del Video

