

---

# Optimización de un sistema de procesamiento de imágenes para el reconocimiento de variables en la pantalla del sistema de Brainlab de HUMANA

---

Santiago Sebastián Galicia Reyes



UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



**Optimización de un sistema de procesamiento de imágenes  
para el reconocimiento de variables en la pantalla del sistema  
de Brainlab de HUMANA**

Trabajo de graduación presentado por Santiago Sebastián Galicia Reyes  
para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2022

Vo.Bo.:

(f) \_\_\_\_\_  
Dr. Luis Alberto Rivera Estrada

Tribunal Examinador:

(f) \_\_\_\_\_  
Dr. Luis Alberto Rivera Estrada

(f) \_\_\_\_\_  
Dr. Juan Carlos Lara

(f) \_\_\_\_\_  
M. Sc. Carlos Esquit

Fecha de aprobación: Guatemala, 08 de Diciembre de 2022.

---

## Prefacio

---

Desde que tengo conciencia me ha interesado la manera en la que funcionan todas las maquinas, me cautivaba completamente el entender como funcionaban y como realizaban todo tipo de movimientos. Con los conocimientos aprendidos desde pequeño y particularmente en la universidad, he desarrollado las habilidades de programación y diseño que se ven plasmados en este trabajo de graduación.

Quiero agradecer a mi hermano, quien siempre se mantuvo conmigo y me ayudó en distintas situaciones a desarrollarme como persona y a cumplir metas. De la misma manera quiero agradecer a mi papá y a mi mamá quienes siempre me motivaron a continuar por mas complicado que pareciera la situación y que siempre se preocuparon por mi bienestar y porque tuviera más de lo que necesité durante toda mi vida.

---

## Índice

---

<b>Prefacio</b>	<b>III</b>
<b>Lista de figuras</b>	<b>VII</b>
<b>Lista de cuadros</b>	<b>VIII</b>
<b>Resumen</b>	<b>IX</b>
<b>Abstract</b>	<b>X</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Antecedentes</b>	<b>2</b>
2.1. Brainlab . . . . .	2
2.2. Investigaciones externas a la Universidad del Valle de Guatemala . . . . .	3
2.3. Investigaciones en Universidad del Valle de Guatemala . . . . .	4
<b>3. Justificación</b>	<b>6</b>
<b>4. Objetivos</b>	<b>7</b>
4.1. Objetivo general . . . . .	7
4.2. Objetivos específicos . . . . .	7
<b>5. Alcance</b>	<b>8</b>
<b>6. Marco teórico</b>	<b>9</b>
6.1. Visión por computador . . . . .	9
6.1.1. Reconocimiento óptico de Caracteres . . . . .	9
6.1.2. Píxel . . . . .	10
6.2. Captura digital de fotografías . . . . .	10
6.2.1. Nitidez de imagen . . . . .	10
6.2.2. Acercamiento de imágenes . . . . .	11
6.2.3. Distancia focal . . . . .	11
6.2.4. Ángulo de visión . . . . .	12

6.2.5. Sensibilidad ISO . . . . .	12
6.2.6. Temperatura del color . . . . .	13
6.2.7. Almacenamiento digital de fotografías . . . . .	13
6.3. Procesamiento digital de fotografías . . . . .	13
6.3.1. Filtros . . . . .	13
6.3.2. Umbralización . . . . .	14
6.3.3. Convolución . . . . .	14
6.3.4. Detección de bordes . . . . .	14
6.3.5. Motores de reconocimiento óptico de caracteres . . . . .	16
6.3.6. Correlación y coincidencia de patrones . . . . .	16
6.4. Aprendizaje automático . . . . .	17
6.4.1. Aprendizaje supervisado . . . . .	17
6.4.2. Aprendizaje no supervisado . . . . .	17
6.4.3. Validación . . . . .	18
6.5. JSON . . . . .	18
6.6. Análisis de Elementos Mecánicos . . . . .	18
6.6.1. Esfuerzos . . . . .	18
<b>7. Sistema de visión por computador</b>	<b>20</b>
7.1. Determinación de la junta a calibrar . . . . .	22
7.2. Determinación del Ajuste . . . . .	26
7.2.1. Procesamiento de Imagen . . . . .	26
7.2.2. Reconocimiento de los dígitos . . . . .	28
7.2.3. Reconstrucción de número . . . . .	30
<b>8. Análisis comparativo de sensores ópticos</b>	<b>31</b>
8.1. Calificador por puntos de interés . . . . .	31
8.2. Calificador por extracción de características . . . . .	32
<b>9. Envió de datos</b>	<b>39</b>
<b>10. Anclaje físico de cámara</b>	<b>41</b>
10.1. Definición del problema . . . . .	41
10.2. Primer Prototipo . . . . .	43
10.3. Segundo prototipo . . . . .	45
<b>11. Conclusiones</b>	<b>48</b>
<b>12. Recomendaciones</b>	<b>49</b>
<b>13. Bibliografía</b>	<b>50</b>
<b>14. Anexos</b>	<b>53</b>
14.1. GitHub . . . . .	53

---

## Lista de figuras

---

1.	Ejemplo robot NeuroMate [1]. . . . .	2
2.	Arreglo 2D de pixeles. Cada pixel tiene sus propias coordenadas $X$ y $Y$ [18]. . . . .	10
3.	Diferencia entre acercamientos de imágenes [22]. . . . .	11
4.	Distancia focal y ángulo de visión [24]. . . . .	12
5.	Comparación ISO [26]. . . . .	12
6.	Cavidades de luz en Mosaico de Bayer [28]. . . . .	13
7.	Proceso de convolución, en donde primero se multiplican el filtro Kernel “w” con la región correspondiente en la imagen y finalmente se suman para obtener el valor del píxel [31]. . . . .	14
8.	derivadas parciales relativas al tipo de borde [32]. . . . .	15
9.	Gaussiano y Laplaciano de Gausiano [32]. . . . .	16
10.	JSON como Array[36]. . . . .	18
11.	Diagrama de funcionalidad de algoritmo de detección de variables. . . . .	21
12.	Distintas pantallas de calibración de VarioGuide. . . . .	22
13.	Captura de pantalla. . . . .	22
14.	Recorte de captura de pantalla. . . . .	23
15.	Procesamiento área de interés: Escala de grises. . . . .	23
16.	Procesamiento área de interés: Suavizado. . . . .	24
17.	Procesamiento área de interés: Umbralización. . . . .	24
18.	Procesamiento área de interés: Limpieza. . . . .	25
19.	OCR PyTesseract a prueba. . . . .	25
20.	Detección del elipsoide con valor de ajuste. . . . .	26
21.	Recorte del elipsoide de interés. . . . .	26
22.	Suavizado de elipsoide de interés. . . . .	27
23.	Umbralización de elipsoide de interés. . . . .	27
24.	Generación de contornos y recorte de dígitos. . . . .	27
25.	Primer dígito recortado. . . . .	27
26.	Segundo dígito recortado. . . . .	28
27.	Rendimiento de DNN ocr con 5 épocas. . . . .	29
28.	Rendimiento de DNN ocr con 40 épocas. . . . .	30
29.	Resultado de reconocimiento OCR. . . . .	30

30.	Formato de prueba utilizado para calificación . . . . .	32
31.	Ejemplo de imagen de prueba de detectores de ejes . . . . .	33
32.	Ejemplo de imagen de prueba para umbralización. . . . .	33
33.	Ejemplo de imagen de prueba de transformador de Hough para lineas y círculos	34
34.	Diagrama de funcionalidad del algoritmo calificador de detección de caracte- rísticas. . . . .	35
35.	Resultados de entrenamiento de modelo con parámetro de 5 épocas. . . . .	37
36.	Resultados de entrenamiento de modelo con parámetro de 15 épocas. . . . .	37
37.	Resultados del calificador con cámara c920. . . . .	38
38.	Resultados del calificador con cámara c270. . . . .	38
39.	Resultados del calificador con cámara Enow. . . . .	38
40.	Prueba de envío de variables de calibración en formato serial. . . . .	39
41.	Captura de pantalla de prueba de envío de variables de calibración en formato TCP. . . . .	40
42.	Pantalla del laboratorio de HUMANA. . . . .	42
43.	Ensamblaje CAD de primer prototipo de anclaje de cámara por enfrente. . . .	43
44.	Ensamblaje CAD de primer prototipo de anclaje de cámara por atrás. . . . .	44
45.	Bisagra en diseño de software CAD. . . . .	44
46.	Resultado de prensa C fabricada. . . . .	45
47.	Ensamblaje de prototipo de acople No.2 en Inventor. . . . .	46
48.	Ensamblaje de prototipo de acople No.2 en Inventor posterior. . . . .	46
49.	Ensamblaje de prototipo de acople No.2. . . . .	47

---

Lista de cuadros

---

1.	Tabla de puntos de interés para calificador . . . . .	31
2.	Datos de poder computacional . . . . .	34
3.	Tabla de dimensiones Dell P2419H . . . . .	41

---

## Resumen

---

La visión por computador es una rama importante de la inteligencia artificial en la que se desarrolla la tecnología bio-inspirada para emular en un computador la capacidad visual del ojo humano.

En este trabajo de graduación se diseñó un mecanismo que permita el anclaje de un sensor óptico a la pantalla del laboratorio de HUMANA de manera que sea lo suficientemente capaz de soportar las cargas para las cuales fue diseñado. Este mecanismo presenta la suficiente agilidad para que su aplicación correspondiente en el laboratorio de HUMANA mejore la efectividad y reduzca el tiempo del sistema de calibración VarioGuide. Para futuras implementaciones se diseñó un envío de datos por formato serial que envía los cuatro ajustes de juntas mecánicas que se obtienen del sistema de reconocimiento de variables de VarioGuide.

Se desarrolló un sistema de procesamiento de imágenes para el reconocimiento de variables en una pantalla de computador que permitiese agilizar el proceso de las operaciones en el laboratorio de HUMANA. Para utilizar este sistema se desarrollo un algoritmo calificador para cámaras en la que se compararon 3 sensores ópticos y devolvió una calificación con la que se verificó el mejor funcionamiento y la cámara mas óptima para la lectura de características que necesita el sistema de reconocimiento de variables.

---

## Abstract

---

Computer vision is an important branch from AI that develops bio inspired technology to be able to emulate the visual capacity from the human eye.

In this graduation work, a mechanism is proposed, this mechanism allows the anchoring of an optical sensor to the HUMANA laboratory screen so that it is sufficiently capable of supporting the loads for which it was designed. This mechanism presents sufficient agility for its corresponding application in the human laboratory to improve effectiveness and reduce the time of the VarioGuide calibration system. For future implementations, relevant data is sent by serial, the format designed sends the 4 mechanical joint settings obtained from the variable recognition system.

An image processing system was developed for the recognition of variables on a computer screen that would speed up the process of operations in the HUMANA laboratory. To use this system, a qualifying algorithm was developed for cameras in which 3 optical sensors were compared and returned a rating that verified the best performance and the most optimal camera for reading the characteristics that the variable recognition system needs.

# CAPÍTULO 1

---

## Introducción

---

En este trabajo se presenta la optimización de un sistema de reconocimiento y procesamiento de imágenes para la lectura adecuada de variables de calibración de la pantalla del laboratorio del Centro de Epilepsia y Neurocirugía Funcional HUMANA. Estas variables se presentan por medio de un sistema de calibración de Brainlab llamado VarioGuide.

Para la optimización del sistema previamente existente se buscó comparar distintos tipos de sensores ópticos mediante métricas que permitieran calificar adecuadamente los mismos y tener una rúbrica mediante la cual se pudo elegir el mejor sensor óptico para la aplicación.

Se diseñó un mecanismo de anclaje para el sensor óptico para comprobar el sistema lectura de variables. Este mecanismo pretende dar facilidad de funcionamiento y agilidad en los procesos quirúrgicos de HUMANA.

Este documento comienza dando antecedentes de la línea de investigación de la Universidad del Valle de Guatemala y también trabajos de investigación que han sido realizados por otras instituciones. El documento también incluye información importante para la comprensión de la investigación dentro del marco teórico y comprende aspectos desde fotografía digital hasta aprendizaje automático.

Se tienen cuatro capítulos en el documento los cuales describen la investigación que se realizó y los resultados logrados. Primeramente se tiene el capítulo principal el cual describe el sistema y seguido se tienen los capítulos auxiliares que describen los trabajos realizados para alcanzar los objetivos específicos.

# CAPÍTULO 2

---

## Antecedentes

---

Los sistemas robóticos y de inteligencia artificial llevan muchos años en estudio y producción. En la Universidad del Valle se han concretado variados trabajos de graduación que buscan unir la robótica y la inteligencia artificial. Existen aplicaciones con estudios externos a la universidad que aunque no llevan el mismo fin que este trabajo, si mencionan aspectos importantes de la unión de la robótica y la inteligencia artificial para diversas aplicaciones que reflejan una perspectiva distinta a la tomada dentro de la Universidad del Valle de Guatemala. Un producto ejemplo en el que la exactitud de la robótica y la asistencia de la inteligencia artificial es esencial lo es el robot NeuroMate [1] el cual se puede visualizar en la Figura 1.



Figura 1: Ejemplo robot NeuroMate [1].

### 2.1. Brainlab

Brainlab es una empresa de software. Brainlab crea e integra productos para distintas fases y etapas del proceso quirúrgico con aplicaciones informáticas desde el posicionamiento del paciente en tratamientos de radio-cirugía o navegación quirúrgica guiada por el software hasta soluciones basadas en la nube diseñadas para garantizar el intercambio seguro de

imágenes [2]. En HUMANA se utiliza el sistema de calibración VarioGuide de Brainlab para corregir y alinear trayectorias planeadas previamente.

## 2.2. Investigaciones externas a la Universidad del Valle de Guatemala

La visión por computador es una rama de la inteligencia artificial de mucho interés para la investigación dado que tiene una alta variedad de aplicaciones por utilizarse. Como puede ser visto en los siguientes ejemplos de trabajos de investigación, el desarrollo de esta tecnología influye en ramas tanto comerciales como científicos.

En la tesis de licenciatura de J. González [3] se desarrolló un sistema de reconocimiento de objetos basado en los algoritmos “FLAN” y “SIFT” con la ayuda de un Raspberry Pi 2. Obtuvo resultados aceptables en la comparación de imágenes precargadas en el sistema para la clasificación y reconocimiento de objetos en una tienda de artículos ferreteros y de pinturas. Haciendo uso de librería de OpenCV y programas de Python logrando reconocer el 100% de los objetos comparados, 75% de estos con procesamiento normal a un 80% de coincidencia y el resto con la ayuda un pre procesamiento añadido adelante en la investigación.

En el trabajo de tesis para licenciatura de G. Viera [4] se desarrollan conceptos básicos de la visión por artificial, los lenguajes utilizados fue Matlab con librerías de visión por computador de OpenCV y Qt Creator con librería de OpenCV. Se presenta así mismo un prototipo de clasificador por características exteriores del grano de cacao orgánico como lo es el tamaño del grano en su etapa final de secado por medio de segmentación de interés en el procesamiento de imágenes. Este trabajo tuvo un resultado del sistema implementado para la clasificación del cacao según tamaño de 88.4 % de acierto en clasificación con una cámara Microsoft® LifeCam Studio™ q2f-00009. Es de recalcar que este trabajo utilizo una metodología mas profunda para tener un sistema controlado en las pruebas utilizando un entorno cerrado con iluminación LED distribuido. Un trabajo de similar aplicación puede ser encontrado en [5] el cual concluye en etapas de procesamiento de señales para la identificación de patrones no deseados en alimentos utilizando el lenguaje de programación Python en su versión 3.7 concluyendo sus resultados el autor indica que la información visual obtenida mediante el registro y procesamiento de imágenes puede vincularse con varios aspectos importantes de la calidad de productos alimenticios.

La aplicación desarrollada en el trabajo final de grado para obtener el grado de Ingeniería Informática de A. Oliva [6] presenta el desarrollo de una herramienta que asiste a la creación de modelos de aprendizaje profundo. La idea de la aplicación desarrollada por Oliva es que los resultados puedan ser empleados para la puesta en práctica en problemas enfocados a la visión por computador dadas las alta fidelidad de reconocimiento. A. Oliva plantea distintos métodos (LBP, Filtros en cascada, HOG) de detección de objetos en la visión por computador detallando las implementaciones y los resultados obtenidos. El trabajo presenta un método de clasificación de patrones de imágenes por medio de aprendizaje profundo de inteligencia artificial, finalmente el autor menciona que una base de datos (como lo ofrecen Google o Microsoft) mas amplia haría mas eficiente el trabajo.

En la tesis de M. Forero y J. Henao [7] se encuentran distintos filtros y transformaciones en procesamiento de imágenes para determinar un sistema de control de calidad automatizado sobre la rugosidad superficial con un análisis en frecuencia de la imagen. Los autores finalizan determinando la confiabilidad del método aplicado con certeza de el 83.11 % en la muestra tomada de 500 pruebas utilizando la cámara Genie HM640 de Teledyne Dalsa. Es de remarcar que en este trabajo se menciona una forma de corrección de un fenómeno (Distorsión óptica), el cual ocurre en cualquier sensor óptico. La corrección del fenómeno les permitió a los autores M. Forero y J. Henao como resultado una calibración mas adecuada y una disminución considerable en las incertidumbres de la toma de imágenes. Dentro del procesamiento de imágenes los autores concluyeron que utilizando el método de transformada de Fourier en una imagen binarizada permitía una reducción considerable de los errores en la medición.

### 2.3. Investigaciones en Universidad del Valle de Guatemala

Según la OPS [8] la epilepsia es uno de los trastornos neurológicos mas comunes llegando a afectar a mas de 50 millones de personas a través del mundo. Un 70 % de las personas afectadas pueden tratarse con fármacos básicos, y solo alrededor de un 10 % requerirán un abordaje especial con dieta o con cirugía. En Guatemala existe el Centro de Epilepsia y Neurocirugía Funcional (HUMANA). En HUMANA se han llevado diversos procedimientos de tratamiento de epilepsia desde el 2006. Gracias al apoyo de HUMANA se han llevado a cabo una serie de trabajos de investigación en la UVG [9].

La Universidad del Valle de Guatemala ha mantenido una relación de investigación con HUMANA y se han desarrollado una serie de trabajos de graduación que han implementado nuevas tecnologías con el fin de mejorar aspectos del laboratorio de HUMANA.

En el trabajo de graduación de José Guerra [10] se implemento un algoritmo de visión por medio de computador utilizando la programación orientada a objetos y multihilos. Este trabajo es continuación de un desarrollo del algoritmo en el lenguaje de programación C++. El software desarrollado por José Guerra [10] tiene como objetivo principal poder reconocer poses robóticas y fue implementado en el lenguaje de programación Python. El limitante de este proyecto fue la falta de pruebas ideales en la mesa Robotat ubicada en la Universidad del Valle, aún así José Guerra desarrolló un prototipo de mesa de prueba en las que dieron lugar las pruebas presentadas en el trabajo. Se tuvo como resultado que la implementación del algoritmo en Python Multihilo fue menos eficiente que la versión de implementación del algoritmo utilizando Python sin hilos. El autor recomendó verificar el paralelismo utilizado en el algoritmo de Python ya que este únicamente utilizaba un núcleo del procesador del computador por lo que los hilos disponibles eran menores en comparación a paralelismo real que utiliza los hilos disponibles en todo el procesador.

El trabajo de graduación de José Ramírez [11] es una continuación de la herramienta de visión por computador de José Pablo Guerra y su enfoque principal es la detección de obstáculos, transmisión de datos entre dispositivos. Se desarrollo una interfaz gráfica que facilita al usuario el uso de la herramienta y se desarrollo una exploración de viabilidad de realizar la paralelización en los códigos para la mejora en rendimiento. En este trabajo se logró la detección de obstáculos y la transmisión de datos entre dispositivos utilizando

protocolos de red UDP y TCP. El limitante principal de este trabajo fue el ser muy específicos de la detección de poses y trayectorias de robots y no de otros objetos de obstrucción. Otra limitante fue la calibración errática de identificadores visuales. Algo que indicó el autor del trabajo fue que la detección de obstáculos es estrictamente funcional cuando son estáticos e invariantes, ya que el algoritmo implementado no soporta un sistema dinámico de detección.

En el trabajo de graduación de Juan González [12] se dieron resultados de un diseño y método de implementación un sistema de reconocimiento óptico de caracteres enfocado a obtener datos desde el sistema de VarioGuide de Brainlab [13]. Los datos obtenidos en el sistema se envían a un sistema de control el cual permite automatizar el sistema mecánico para optimizar el tiempo de cirugías. Este trabajo es parte de 4 etapas en el Proyecto del brazo de HUMANA divididos en etapas de software y diseño mecánico, el sistema OCR de Juan González cuenta con 7 etapas de procesamiento y obtuvo un reconocimiento dentro de los márgenes de error de Brainlab utilizando dos tipos de sensores ópticos, una cámara Logitech 1080p y una cámara GoPro Hero 8, con esta última se obtuvieron mejores resultados debido a la mejor resolución del sensor. El autor recomienda realizar un nuevo diseño del brazo que sostiene la cámara de reconocimiento para que la calibración sea más simple y el sistema sea más preciso. También se indica un problema con el algoritmo de inversión y filtrado de colores y obtención de píxeles en el algoritmo utilizado de reconocimiento. El sistema óptico de reconocimiento de caracteres y automatización del sistema mecánico no fue completamente implementado al brazo HUMANA ya que el diseño no es exacto y las pocas pruebas realizadas fueron erráticas.

## CAPÍTULO 3

---

### Justificación

---

En el trabajo de graduación para el grado de licenciatura de J. González [12] se logró un avance significativo en el software de reconocimiento a utilizar con VarioGuide de Brainlab. Sin embargo los resultados obtenidos no fueron totalmente concluyentes y del sistema construido no se realizaron pruebas en un sistema cerrado o pruebas con mas sensores ópticos. El sistema de González presenta un acople del sensor óptico rígido y difícil de aplicar para el brazo biomédico de HUMANA.

Existen diversos algoritmos que no fueron implementados en el sistema de procesamiento de imágenes de González, los cuales pueden ayudar a optimizar el funcionamiento y desempeño del mismo. En particular, se pudo mejorar la calibración del mecanismo de HUMANA. Por lo anterior mencionado, en este trabajo se planteó la necesidad de cumplir con los requisitos de exactitud y de confiabilidad necesarios para la implementación adecuada al brazo biomédico HUMANA. Así mismo, se necesitó la confiabilidad de que el sensor funcione de manera correcta cuando sea necesario calibrar la máquina por medio de el software VarioGuide de Brainlab.

Con este trabajo se logró obtener una comparación entre sensores ópticos según la habilidad de reconocer características. Se implementó una nueva aproximación del mecanismo de acople de sensor óptico. Así mismo se optimizó los sistemas de reconocimiento de variables previamente desarrollados.

# CAPÍTULO 4

---

## Objetivos

---

### 4.1. Objetivo general

Optimizar el sistema de reconocimiento y procesamiento de imágenes y caracteres con la pantalla del sistema de Brainlab del Centro de Epilepsia y Neurocirugía Funcional HUMANA para que lea de forma adecuada las variables de calibración.

### 4.2. Objetivos específicos

- Comparar la eficiencia de distintos tipos de sensores ópticos y cámaras para tareas de reconocimiento y procesamiento de imágenes.
- Implementar un sistema de procesamiento de imágenes capaz de identificar la junta que está siendo calibrada en el mecanismo y el error desplegado.
- Implementar un protocolo de comunicación de envío de datos al sistema de control del mecanismo del brazo biomédico de HUMANA.
- Diseñar y fabricar un mecanismo de acople para un sensor óptico a la pantalla de Brainlab de HUMANA.

## CAPÍTULO 5

---

### Alcance

---

Para el sistema de comparación de cámaras se limitó a probar 3 distintas cámaras debido al alto costo que puede representar la compra de mas cámaras o sensores ópticos. Se logró enviar por medio de un formato estandarizado los valores a calibrar desplegados por VarioGuide, se limito a verificar únicamente el envío y recepción en un microcontrolador Arduino ya que se espera que en otra fase se involucre el manejo de datos para hacer de manera automatizado el proceso de calibración.

Para este proyecto se tiene un alcance limitado principalmente por la dificultad de obtener acceso a las instalaciones del laboratorio de HUMANA para pruebas adecuadas tanto del mecanismo de acople como de los algoritmos de reconocimiento de variables de la pantalla.

# CAPÍTULO 6

---

Marco teórico

---

## 6.1. Visión por computador

La visión por computador se refiere a el uso de inteligencia artificial para analizar, comprender y responder a imágenes digitales o videos. Al aplicar algoritmos de inteligencia artificial a la entrada de cámaras permite que los datos se procesen y se evalúen para reconocer patrones. Analizando una serie de imágenes, un programa puede ser entrenado para reconocer, clasificar y reaccionar desde la información que recibe. Mientras que la visión por computador puede ser entrenada para resolver casi cualquier problema visual, cada modelo es extremadamente estrecho en sus capacidades. Solamente puede reconocer lo que se le ha enseñado a reconocer. La visión por computador puede confiablemente ejecutar la misma tarea una y otra vez sin la problemática del error humano causado por fatiga o por distracciones. Los modelos pueden incluso ser entrenados para procesar información a altas velocidades o en espectros visuales distintos a los visibles por el ojo humano, como luz ultra violeta o luz infrarroja [14].

### 6.1.1. Reconocimiento óptico de Caracteres

Los sistemas de reconocimiento óptico de caracteres (*Optical character recognition - OCR*) son sistemas que puede leer datos escritos reconociéndolos a altas velocidades, un carácter a la vez. La gran mayoría de sistemas OCR aún no pueden leer caracteres o palabras escritas a mano o documentos con degradación física. Actualmente los sistemas OCR pasan la toma de una imagen por varios procesos en los que se organiza la imagen para obtener los caracteres, se extraen los caracteres y se clasifican los caracteres extraídos [15]. Un ejemplo de uso común es Live Text incluido en el sistema operativo de Apple para sus teléfonos iOS 15 [16].

### 6.1.2. Píxel

Un píxel es un pequeño recuadro de color, al juntar varios de estos píxeles en un arreglo de filas y columnas se puede formar una imagen digital. Un píxel contiene la información digital del color RGB. Píxel proviene de la combinación PIC y EL de *Picture-Element* en inglés. El píxel es la unidad básica de una imagen digital ya que esta ultima tendrá el tamaño de X columnas de píxeles por Y filas de píxeles de alto. Un método de almacenaje de imágenes es el mapa de bits, la información se guarda como una secuencia de números definiendo el color de cada píxel [17].

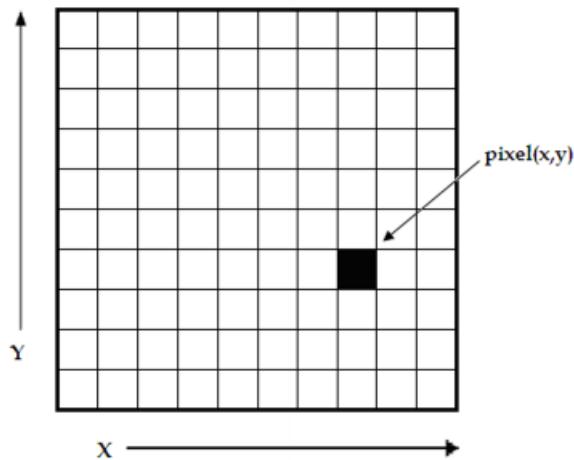


Figura 2: Arreglo 2D de píxeles. Cada pixel tiene sus propias coordenadas  $X$  y  $Y$  [18].

## 6.2. Captura digital de fotografías

### 6.2.1. Nitidez de imagen

La nitidez de una imagen es el grado de claridad del detalle en una imagen. Existen formas digitales para mejorar la nitidez en una imagen, como por ejemplo aplicar algunos filtros, pero la nitidez siempre estará directamente conectada con las capacidades de la cámara y la toma de la fotografía. Los principales factores que afectan a la nitidez de una imagen son:

#### Acutancia

Contraste de luminosidad entre los bordes de los detalles y sus fondos [19].

## Resolución

La resolución de una fotografía digital es su cantidad de píxeles por unidad de longitud, es decir, es la relación entre el tamaño en píxeles de una imagen y sus dimensiones físicas, que se manifiesta en un dispositivo de salida como una pantalla. Una resolución mayor implica más píxeles por unidad de longitud, lo que quiere decir que el nivel de detalle es mayor. La resolución describe la habilidad que tiene una cámara para distinguir entre el espacio de los detalles en los elementos [20].

### 6.2.2. Acercamiento de imágenes

El acercamiento de imágenes mas conocido como *zoom* es el aumento de la imagen, acercar la imagen sin mover el punto de toma. Para realizar este aumento existen dos métodos [21]. Una comparación visible es dada en la Figura 3.

El *zoom* digital explicado de forma sencilla es tomar un trozo del sensor fotográfico de la cámara y luego estirar el resultado colocando píxeles intermedios. Un ejemplo simple es que si se tiene un sensor de  $16 \text{ píxeles} \times 16 \text{ píxeles}$ , hacer el *zoom* digital es tomar únicamente  $8 \text{ píxeles} \times 8 \text{ píxeles}$  y llenar digitalmente el resto con una mezcla de colores de manera que el resultado parezca agradable a la vista del usuario.

Una cámara con *zoom* óptico acerca la imagen antes de que sea capturada. La cámara hace esto moviendo diferentes partes de la lente entre sí. Esto acerca la imagen sin perder calidad. Algunos lentes pierden una ligera cantidad de contraste y enfoque [21].



Figura 3: Diferencia entre acercamientos de imágenes [22].

### 6.2.3. Distancia focal

Esta es una medida por lo general en milímetros que indica la distancia entre el sensor y el centro de un lente óptico. Mientras esta medida sea más grande, más estrecha será el ángulo de visión.[23]

#### 6.2.4. Ángulo de visión

El ángulo de visión es la amplitud de la escena que es capaz de proyectar sobre el plano focal. El ángulo de visión se extrae de la siguiente ecuación:

$$\alpha = 2 \times \arctan\left(\frac{d}{2 \times f}\right) \quad (1)$$

En la ecuación 1 se utiliza "d" como el tamaño del sensor en la dirección que se pretende obtener el ángulo y "f" como la distancia focal al lente [24].

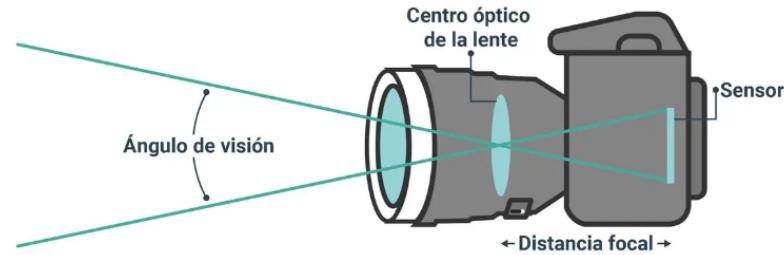


Figura 4: Distancia focal y ángulo de visión [24].

#### 6.2.5. Sensibilidad ISO

Esta sensibilidad es una escala que determina el grado de sensibilidad del sensor de una cámara digital respecto a la luz que se proyecta sobre el sensor. Mientras más grande el número de ISO es mayor la capacidad de recibir luz. En conjunto con la apertura del diafragma y el tiempo de exposición se definen la exposición (cantidad de luz capturada) en las fotografías [25].

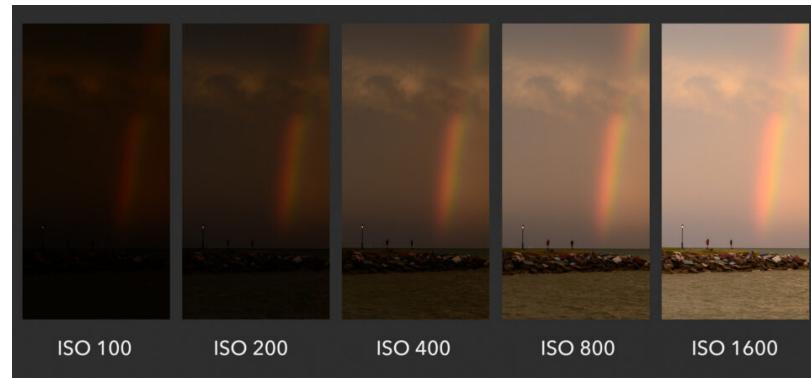


Figura 5: Comparación ISO [26].

### 6.2.6. Temperatura del color

La temperatura de color de una fuente de luz es la temperatura de un radiador ideal de cuerpo negro que irradia luz de tono comparable al de la luz fuente. La temperatura de color se expresa convencionalmente en la unidad de valor absoluto de temperatura, el Kelvin (K). La temperatura del color es una característica de la luz visible que tiene importantes aplicaciones en iluminación, fotografía, videografía, editorial, manufactura, astrofísica y otros campos [27].

### 6.2.7. Almacenamiento digital de fotografías

#### Mosaico de Bayer

Este filtro es un tipo de matriz de filtros, rojos, verdes y azules, que se sitúan en un sensor digital de imagen para que cada foto diodo reciba la información de luminosidad correspondiente a los colores primarios. Con una interpolación de 4 foto diodos vecinos se recibe un píxel de color. El mosaico de Bayer se forma de un 25 % de filtros rojos 50 % verdes y 25 % azules [28]. Se puede observar el comportamiento de el filtro en la Figura 6.

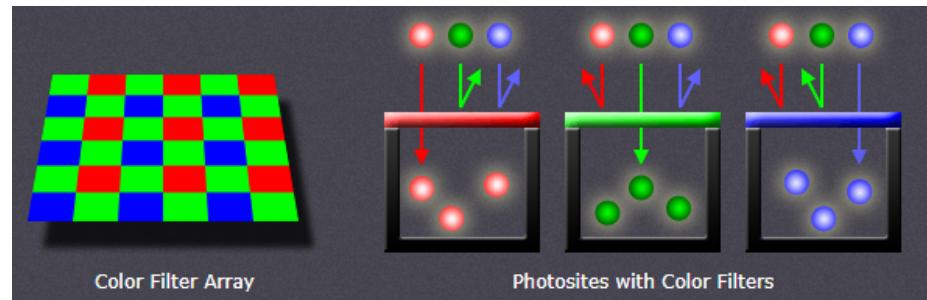


Figura 6: Cavidades de luz en Mosaico de Bayer [28].

## 6.3. Procesamiento digital de fotografías

### 6.3.1. Filtros

El filtrado lineal es un proceso en el que parte del espectro de frecuencias de la señal (imagen) es modificado por la función de transferencia del filtro. En general, los filtros considerados son lineales e invariantes al desplazamiento y, por lo tanto, las imágenes de salida se caracterizan por la suma de convolución entre la imagen de entrada y la respuesta de impulso del filtro [29].

### 6.3.2. Umbralización

La umbralización o *Thresholding* es un método de segmentación de imágenes en la que se separan los objetos del fondo de una imagen[30].

### 6.3.3. Convolución

El método de aplicación de un filtro es comúnmente referido como convolución y esta trabaja en una vecindad de valores representados en matrices [29]. Se puede ver el proceso de convolución en la Figura 7.

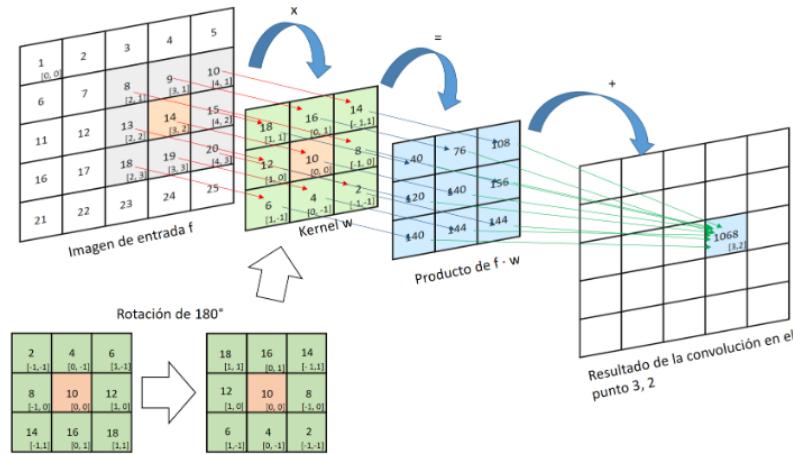


Figura 7: Proceso de convolución, en donde primero se multiplican el filtro Kernel “w” con la región correspondiente en la imagen y finalmente se suman para obtener el valor del píxel [31].

### 6.3.4. Detección de bordes

Los bordes naturalmente tienen una importancia especial en la percepción de objetos. Los bordes de un objeto presentan una ayuda al momento de comprender aspectos del mismo objeto como lo es la textura, la iluminación y la forma. Existen varios métodos de detección de objetos, por ejemplo:

#### Detectores de bordes por medio de gradientes

Los bordes se forman cuando hay un cambio drástico en la iluminación de uno o mas píxeles. Si se considera la imagen como una función, un cambio drástico en su valor es considerado como un borde. Por tanto un píxel es un borde si la magnitud de la primer derivada de la imagen en ese píxel es grande [32].

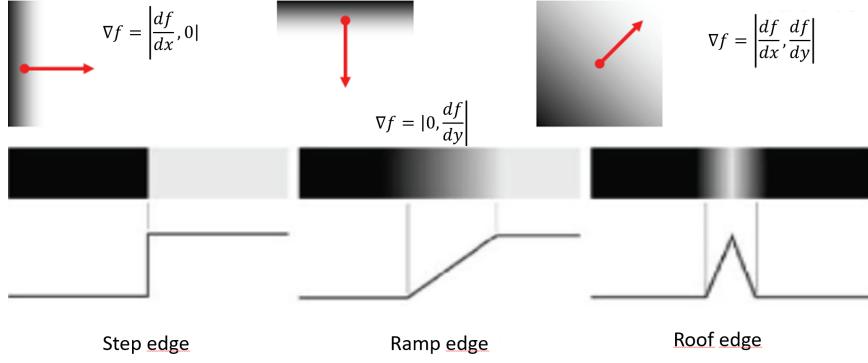


Figura 8: derivadas parciales relativas al tipo de borde [32].

Si se denota la imagen como una función de 2 dimensiones el gradiente de la función tiene dirección [32]. Se puede desarrollar el gradiente en el píxel en dirección  $x$  y en dirección  $y$  separadamente y luego combinarlos para obtener el gradiente de la imagen de la siguiente manera:

$$\nabla f = \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) = (g_x, g_y) \quad (2)$$

### Detectores de bordes basados en curvatura

Estos detectores se basan en que los bordes ocurren cuando el gradiente de la imagen alcanzan un máximo o mínimo. En los puntos de máximos o mínimos del gradiente la segunda derivada o la curvatura tiene un punto cruzando 0. La ventaja de este método es que responde similar a las distintas direcciones de bordes y encuentra la posición correcta del borde. La curvatura en la dirección  $x$  es:

$$g_x - g_{x-1} = f(x+1, y) - 2f(x) + f(x-1, y) \quad (3)$$

Si se consideran las diferencias finitas del gradiente en la dirección vertical y se agregan a la ecuación (3) el operador de la curvatura tomara en cuenta no solo al píxel mismo sino que a los vecinos conectados junto al píxel [32]. Los operadores de segunda derivada son:

- Laplaciano: Encuentra ejes con orientación para afuera y orientación para adentro.

$$\nabla^2 f = \left( \frac{\partial^2 f}{\partial^2 x}, \frac{\partial^2 f}{\partial^2 y} \right) \quad (4)$$

- LOG: Combina el efecto de un suavizamiento gaussiano con el laplaciano en una sola mascara.

$$\nabla^2 G = \left( \frac{\partial^2 G}{\partial^2 x}, \frac{\partial^2 G}{\partial^2 y} \right) \quad (5)$$

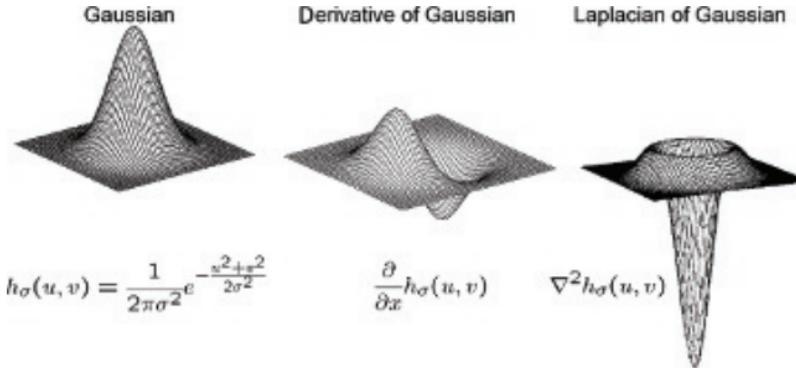


Figura 9: Gaussiano y Laplaciano de Gausiano [32].

### 6.3.5. Motores de reconocimiento óptico de caracteres

Los motores de reconocimiento de caracteres utilizando sensores ópticos permiten reconocer texto y datos eficientemente mediante imágenes de distintas estructuras. Existen diversos motores de reconocimiento, entre estos están:

#### Tesseract

Tesseract es un motor OCR el cual es de acceso publico. Este contiene una librería de procesamiento de imágenes con capacidad de leer gran variedad de formatos de imagen extraer el texto contenido en estas. Tesseract contiene soporte en formato unicode (UTF-8) y puede reconocer mas de 100 lenguajes [33].

#### Cuneiform

Cuneiform es un sistema de reconocimiento de caracteres multilenguaje desarrollado por *Cognitive Technologies*. Contiene paquetes para reconocimiento de formato de texto y análisis de sintaxis. Este motor mantiene la estructura y formato del texto en la imagen.

### 6.3.6. Correlación y coincidencia de patrones

Si  $y_t = f(x_t)$ , una relación entre  $x$  y  $y$  puede observarse. La fuerza de la relación lineal es frecuentemente caracterizada como un coeficiente de correlación entre  $x$  y  $y$ . La convolución cuando se usa en funciones compensadas por su media, se puede usar para encontrar el grado de similitud entre una plantilla y una región en la imagen [29].

## 6.4. Aprendizaje automático

El aprendizaje automático (*Machine learning*) se basa en crear modelos basados principalmente en programas estadísticos. Los modelos toman una entrada y devuelven una salida. El aprendizaje automático utiliza algoritmos que son capaces de aprender de la información que se le provee [34].

Algunas de las tareas que pueden ser resueltas fácilmente con aprendizaje automático son:

- Clasificación: el computador especifica la categoría  $k$  a la que pertenece una entrada.
- Eliminación de ruido: El computador obtiene un ejemplo con información corrupta de manera aleatoria y regresa el ejemplo de manera “ limpia”.
- Sintetización: El computador pretende generar nuevos ejemplos similares a los que se utilizaron para entrenar el modelo.
- Regresión: el computador predice un valor numérico dependiendo la entrada.
- Transcripción: El computador observa una representación abstracta de información y la transcribe a una forma textual.

### Características

Las características son una propiedad medible de un objeto individual. Por lo general son fenómenos en cantidades numéricas [35]. Hay dos tipos de características:

- Cuantitativas: Definen valores como números(*integers, floats, counts, calificaciones, etc.*).
- Cualitativas: Usualmente son etiquetas o símbolos que contienen información útil en formato mas parecido a lo que entienden fácilmente los humanos.

#### 6.4.1. Aprendizaje supervisado

Los algoritmos de aprendizaje supervisado reciben un conjuntos de datos que contienen características acompañados de una etiqueta [34].

#### 6.4.2. Aprendizaje no supervisado

Los algoritmos de aprendizaje no supervisado por lo general reciben un set de datos que contienen varias características, después aprenden las propiedades mas sobresalientes de la estructura de datos y pueden realizar las tareas que le sean solicitadas [34].

### 6.4.3. Validación

Para obtener los mejores parámetros posibles de un modelo y poder afinarlos se necesita un set de validación. Un set de validación es un conjunto de datos, que proviene de los datos ingresados al modelo para entrenarlo. La división de los datos de entrenamiento por lo general se utiliza un 75 % para datos de entrenamiento y 25 % para datos de validación [34].

#### Sobre-ajuste y sub-ajuste

El sobre-ajuste ocurre cuando el algoritmo ha aprendido de mas de el set de datos generando discriminantes y reglas que no deberían de existir.

El sub-ajuste ocurre cuando el modelo del aprendizaje automático no puede capturar la tendencia implícita de los datos. Este sub-ajuste produce un modelo con rendimiento predictivo sumamente deficiente.

Para solucionar este tipo de ajustes se debe de realizar un procesamiento de los datos para introducir aleatoriedad. También es importante comenzar con modelos simples e ir subiendo de complejidad según beneficie [35].

## 6.5. JSON

JSON es un formato de intercambio de datos de bajo peso. Este es un formato que se construye en dos estructuras, Una colección desordenada de valores pares o una lista ordenada de valores [36]. En la Figura 10 se observa el formato para un array.

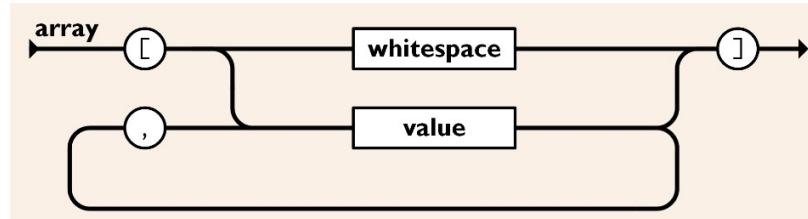


Figura 10: JSON como Array[36].

## 6.6. Análisis de Elementos Mecánicos

### 6.6.1. Esfuerzos

Los esfuerzos son relaciones entre la fuerza aplicada y el área de la sección transversal donde esta actuando una fuerza aplicada en otras palabras son reacciones de los elementos de estructuras para nos ser deformado por cargas. Los esfuerzos normales tienen nomenclatura

$\sigma$  mientras que los esfuerzos cortantes tienen nomenclatura  $\tau$ . Las unidades de esfuerzo en el sistema internacional se mide en  $1\frac{N}{m^2}$  que equivale a 1 *Pascal(Pa)* [37]

### Esfuerzos por flexión

Son aqueos que llevan consigo mismo la cantidad de deformación que pueda sostener un elemento, ya que las cargas que pueda sostener un elemento producen momentos de flexión flexionando los elementos.

- Esfuerzos normales para vigas por flexión: Ocurre en vigas que soportan cargas con direcciones normales a la superficie de la viga generando en los soportes una flexión. Su valor se puede calcular utilizando la siguiente ecuación:

$$\sigma = \frac{M \times y}{I} \quad (6)$$

- Esfuerzos cortantes para vigas por flexión: Este tipo de esfuerzo ocurre en vigas que soportan cargas transversales. Es máximo en el eje neutro y es nulo en los extremos de la viga. Su valor se puede calcular utilizando la siguiente ecuación:

$$\tau = \frac{V \times Q}{I \times t} \quad (7)$$

$$\tau_{max} = \frac{3 \times V}{2 \times A} \quad (8)$$

## CAPÍTULO 7

---

### Sistema de visión por computador

---

El sistema de visión por computador cuenta con el comportamiento de forma secuencial. Este algoritmo utiliza la librería de Tesseract y una red neuronal de aprendizaje automático para realizar el procedimiento de recolección de variables. En la Figura 11 se describe el proceso por el que pasa la captura de la pantalla del laboratorio de HUMANA. Inicia reconociendo el entorno por medio del sensor óptico utilizado, cuando el sistema reconoce que puede obtener una imagen clara de la pantalla de computador toma la captura. El sistema ingresa la captura de la pantalla a un procesamiento de imágenes en el que a la imagen se le coloca un umbral para resaltar los caracteres y que para la librería de Tesseract y la red neuronal reconozca con menor perdida los caracteres.

Una de las etapas mas importantes es cuando se procesan los datos que recolectó el sistema por medio de la librería de Tesseract ya que esto será la guía para reconocer la junta a manipular. En esta etapa se procesan los datos para determinar tanto la junta que se está calibrando, como el porcentaje de error de calibración que reconoce VarioGuide.

El procesamiento toma como pilar principal la calidad de imagen del sensor óptico elegido y condiciones de pre-procesamiento. Por esto que los parámetros de la umbralización fueron afinados utilizando pruebas en Python con la cámara Logitech C920s.

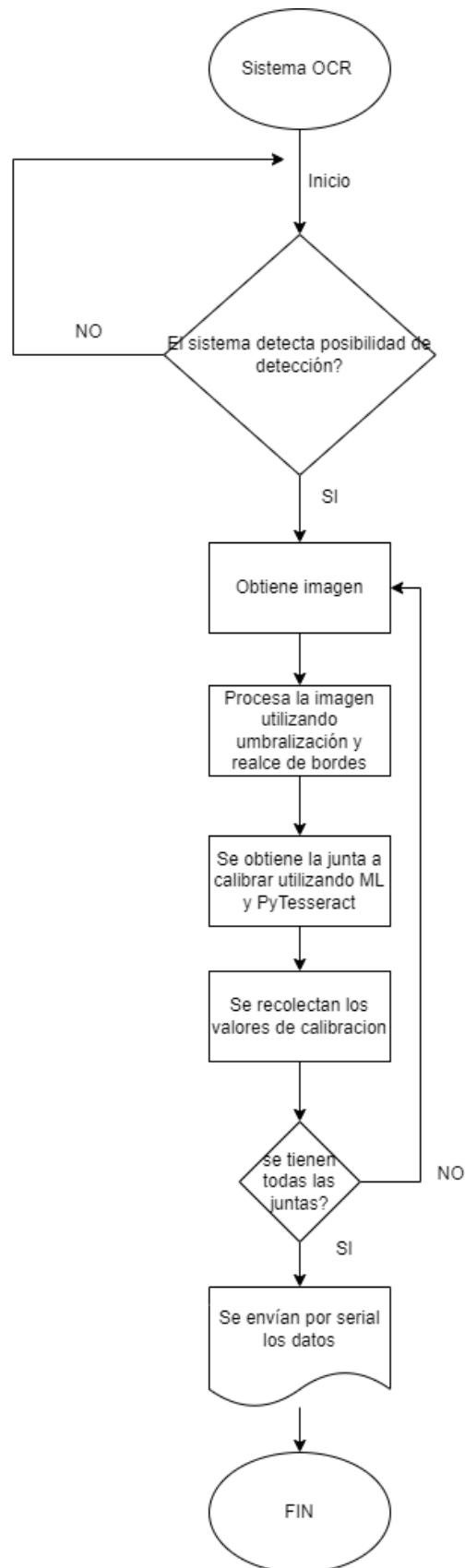


Figura 11: Diagrama de funcionalidad de algoritmo de detección de variables.

Las distintas pantallas desplegadas por el sistema VarioGuide de Brainlab en el laboratorio de HUMANA con la información de la junta a leer y su grado de calibración se encuentran en la Figura 12. Estas pantallas fueron las utilizadas para probar el sistema de reconocimiento de variables durante todo el proceso de desarrollo y de pruebas.



Figura 12: Distintas pantallas de calibración de VarioGuide.

## 7.1. Determinación de la junta a calibrar

El sistema de procesamiento comienza con una función para determinar el foco de imagen y la resolución de la cámara. En este caso se utilizó un ISO por defecto en un rango de 250 a 2047 ya que este valor no se puede forzar en la cámara web. Se utilizó un foco manual de 30 y se utilizó una resolución de  $1920 \times 1080$ . En la Figura 13 se observa la captura que obtiene esta función.



Figura 13: Captura de pantalla.

Después de obtener la imagen, esta se procesa utilizando una función de OpenCv llamada “cvtColor” con la que se vuelve formato BGR. Esta imagen BGR se recorta con una función que utiliza el centro de la imagen en las coordenadas de píxeles ( $x = 960$ ,  $y = 540$ ) con las que elimina las franjas de color negro de los laterales izquierdo y derecho y de el superior e inferior. En la Figura 14 se observa el procesamiento de recorte.

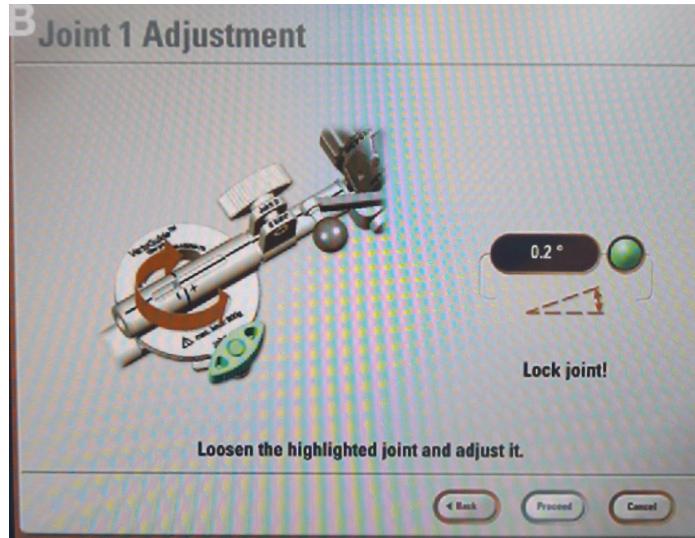


Figura 14: Recorte de captura de pantalla.

Cuando el programa ya consiguió satisfactoriamente recortar el área de interés, comienza el procesamiento de esta imagen para reconocimiento y determinación de la junta manipulada. Se trabajó una función para convertir a escala de grises la imagen utilizando OpenCv y su función “cvtColor” con la opción “BRG2GRAY” que permite a la imagen adoptar el formato blanco y negro como se observa en la Figura 15.

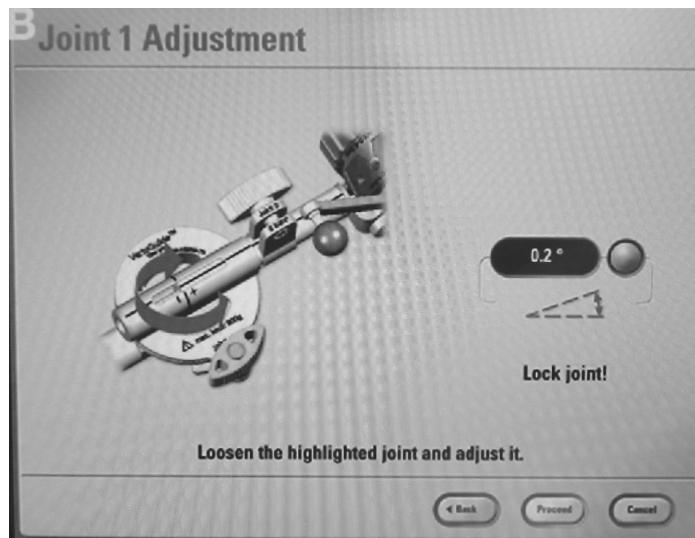


Figura 15: Procesamiento área de interés: Escala de grises.

Con el propósito de ayudar el motor PyTesseract se utiliza un suavizado utilizando suavizado Gausiano con un kernel de  $5 \times 5$ . La imagen suavizada es mostrada en la Figura 16. Después se utiliza una etapa de umbralización utilizando umbral binario, dado que el motor Tesseract lee dígitos negros sobre fondo blanco, en combinación del sistema “Otsu” de umbralización automática. El sistema de Otsu procesa el histograma de distribución de píxeles de la imagen, calcula el valor de umbral óptimo por medio de iteraciones y reemplaza píxeles con blanco en lugares donde la saturación es mayor que el umbral óptimo y con píxeles negros en lugares donde la saturación es menor que el umbral óptimo.

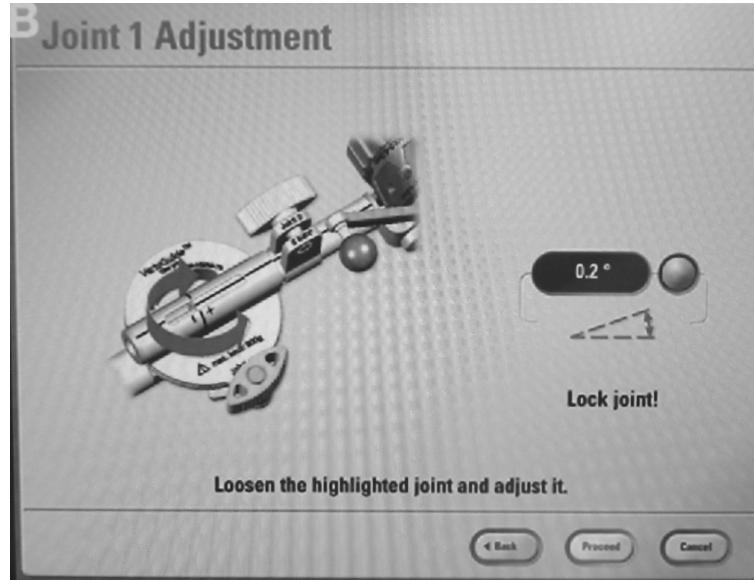


Figura 16: Procesamiento área de interés: Suavizado.

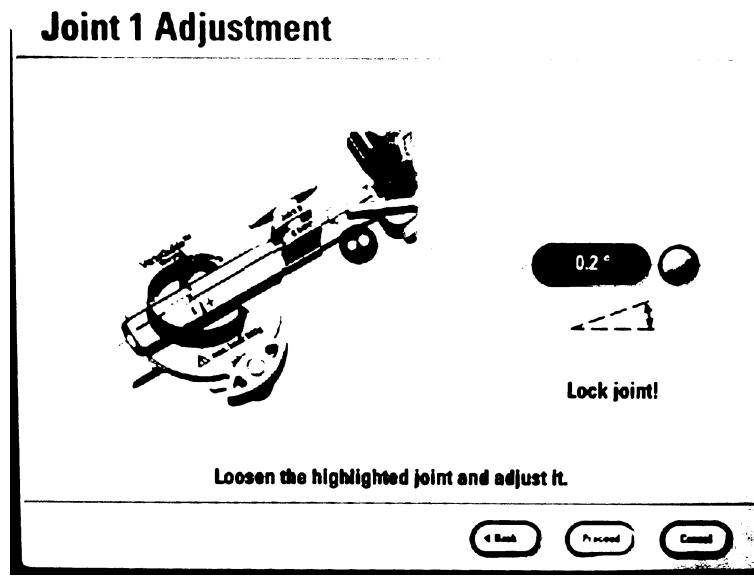


Figura 17: Procesamiento área de interés: Umbralización.

Para eliminar el ruido que pueda causar la operación de umbralización de Otsu se generó una función. Esta función utiliza operaciones morfológicas para eliminar puntos negros que puedan afectar la lectura OCR. Las operaciones que utiliza esta función de eliminación de ruido son:

- Dilatación: Esta operación convierte los píxeles de fondo negro que rodean un objetivo se vuelvan blancos, en otras palabras, el objetivo “se vuelve mas gordo”.
- Erosión: Esta operación convierte los píxeles en la orilla de un objetivo se conviertan negros, por lo tanto, el objetivo “se vuelve mas delgado”.
- Cerrado: Esta operación elimina puntos o píxeles negros dentro de los objetivos en primer plano.

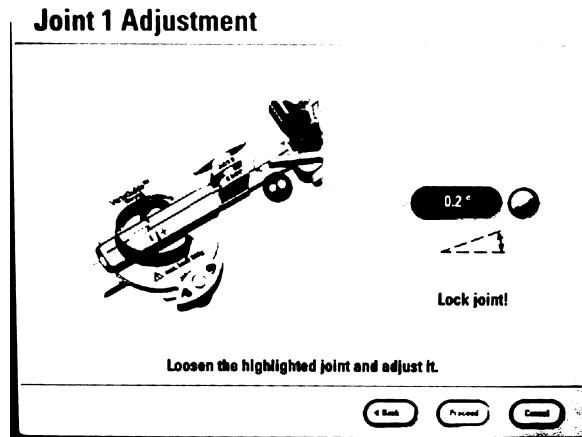


Figura 18: Procesamiento área de interés: Limpieza.

Finalmente después del procesamiento de limpieza en la Figura 18 se realiza el escaneo de PyTesseract con la función “`pytesseract.image_to_string()`” en la configuración “11” que permite la detección de la máxima cantidad de caracteres sin orden en particular. Se generó una función que maneja la información obtenida por OCR. Esta función utiliza “`find()`” para encontrar el tipo de junta que se tiene. En la Figura 19 se observa el string generado por PyTesseract.

```
Proto_OCR_V03 ×
↑ C:\Users\galic\PycharmProjects\UVG\Prototipos\Scripts\python.exe "
↓ Joint 1 Adjustment
Epo
Pet
Lock joint!
Loosen the highlighted joint and adjust it.

Oe @e-
```

Figura 19: OCR PyTesseract a prueba.

## 7.2. Determinación del Ajuste

### 7.2.1. Procesamiento de Imagen

En esta etapa se regresa al recorte del área de interés de calibración. en esta etapa se utiliza la imagen recortada y se convierte a RGB utilizando “cvtColor(BGR2RGB)” de openCv. Se generó una función aparte con la que se genera una mascara manual que busca el color de interés de la sección elipsoidal que contiene el ajuste de la junta. Esta mascara deberá ser ajustada a los colores que se detecte en la pantalla del laboratorio de HUMANA, generalmente estos colores están en el rango de 35 a 75 de 255 en RGB. Luego de generarse esta mascara, la función generada implementa la función de OpenCv “FindContours” para redondear únicamente las secciones de la imagen que contengan colores permitidos por la mascara de color. La función busca un contorno de área de píxeles mayor a 5000 y determina el centroide del área. El procesamiento de esta función puede ser vista en la Figura 20.

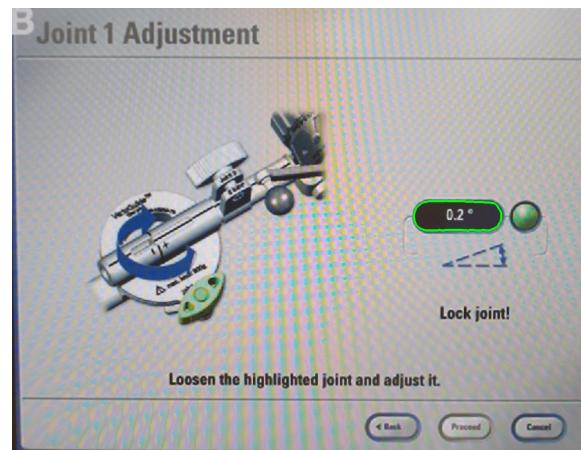


Figura 20: Detección del elipsoide con valor de ajuste.

Con el centroide de la función anterior se genera un recorte a una imagen de resolución  $200 \times 80$  como se puede ver en la Figura 21. A este recorte se le aplica una función de OpenCv para convertirla de RGB a escala de grises en 8 bits. Misma imagen que se suaviza como puede ser observado en la Figura 22



Figura 21: Recorte del elipsoide de interés.



Figura 22: Suavizado de elipsoide de interés.

Se procesa el recorte suavizado utilizando una función de umbralización binaria con umbral automático Otsu como puede ser visto en la Figura 23. Se generó una función que reconoce los contornos de los dígitos con área entre 150 y 600 píxeles<sup>2</sup> de la imagen umbralizada como puede ser visto en la Figura 24. La función guarda una imagen cuadrada por contorno del dígito en el folder del programa, este resultado puede ser visto en la Figura 25 y en la Figura 26.



Figura 23: Umbralización de elipsoide de interés.



Figura 24: Generación de contornos y recorte de dígitos.

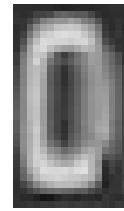


Figura 25: Primer dígito recortado.



Figura 26: Segundo dígito recortado.

Finalmente se diseñó una función para procesar las imágenes de los dígitos convirtiéndolos de formato BGR a formato de escala de grises y haciendo uso de la función de resize de OpenCv, cada dígito se escala a una imagen de resolución  $28 \times 28$ .

### 7.2.2. Reconocimiento de los dígitos

Dado que esta fase es la de mayor interés en precisión, se utilizo un sistema de reconocimiento más robusto. Entre las opciones estaba disponible:

- PyTesseract.
- KerasOCR.
- EasyOCR.
- Red de Machine Learning.

Se descartó EasyOCR por su alta tasa de error (4.7%) y PyTesseract se descartó por qué, aunque es muy preciso, es un motor mas optimizado en el alfabeto y no con dígitos. Ambos motores son altamente dependientes de la resolución de la imagen de entrada [38].

KerasOCR se descartó dada la alta dificultad de manejar la salida de texto reconocido y porque en bajas resoluciones tiende a confundirse entre dígitos.

Se decidió utilizar una red neuronal de aprendizaje automático. Esta red secuencial está generada en Tensorflow Keras. Esta red utiliza una entrada de imágenes de  $28 \text{ píxeles} \times 28 \text{ píxeles}$  y termina en una etapa de 10 neuronas. Utilizando la función de numpy "np.argmax()" el programa elige que dígito se refiere. Las etapas de la red neuronal son:

1. Capa de aplanado con entrada  $28 \times 28$ .
2. Capa densa de 1000 neuronas con función de activación tangente hiperbólica.
3. Capa densa de 500 neuronas con función de activación linear unitaria rectificada.
4. Capa densa de 100 neuronas con función de activación sigmoide.
5. Capa densa de 10 neuronas con función de activación softmax.

Es importante para que la red no tenga sobre ajuste o sub ajuste que las funciones de activación entre capas densas sean variadas, de lo contrario se generaría una alta perdida y probablemente un resultado menos preciso.

La red neuronal se entreno utilizando el set de datos “MNIST” base de Keras. Se utilizo para compilar la red neuronal un optimizador “Adam” con ritmo de aprendizaje de 0.001 para un aprendizaje mas detallado. Se utilizó una función de perdida de tipo entropía categórica esparcida dada que la entrada no esta codificada utilizando “one-hot encoder”, mas bien es entrada bruta. Para este modelo se utilizo una métrica de precisión.

El modelo se construyó en un principio con 5 épocas para ver la eficacia del modelo. Con 5 épocas se obtuvo un desempeño de modelo del 92.057% y una perdida de 0.248, como se observa en la Figura 27. Mejorando el modelo se construyó 40 épocas y se obtuvo un desempeño de modelo del 95.175% y una perdida de 0.187, como se observa en la Figura 28.

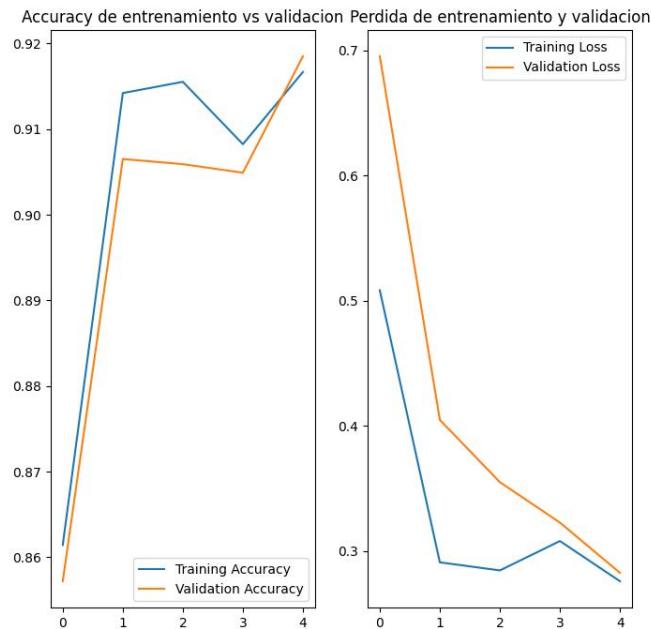


Figura 27: Rendimiento de DNN ocr con 5 épocas.

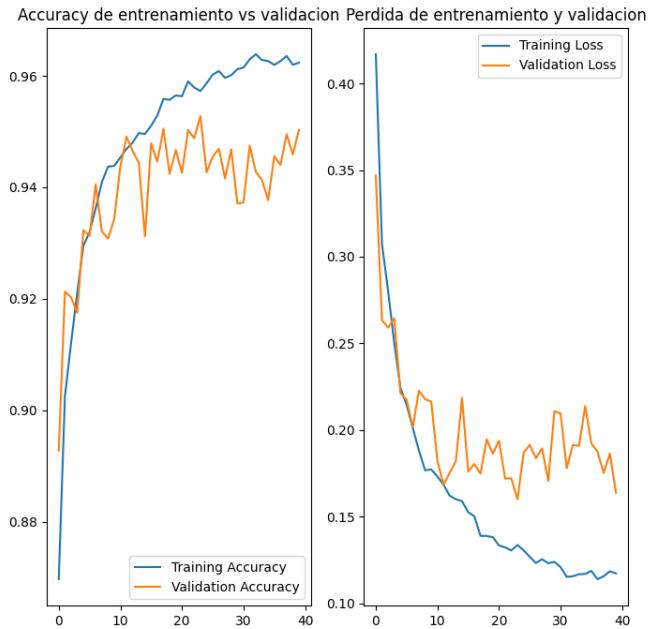


Figura 28: Rendimiento de DNN ocr con 40 épocas.

### 7.2.3. Reconstrucción de número

Finalmente se construyó una función para detectar si la imagen de los dígitos existe en el archivo actual y construye el número utilizando formato “Float”. En la Figura 29 se observa el sistema funcionando adecuadamente en una prueba.

```
Proto_OCR_V03 ×
↑ 2022-11-25 01:43:55.106861: W tensorflow/compiler/xla/stream_executor/platform
↓ 2022-11-25 01:43:55.107031: W tensorflow/core/common_runtime/gpu/gpu_device.cc
Skipping registering GPU devices...
2022-11-25 01:43:55.107573: I tensorflow/core/platform/cpu_feature_guard.cc:19
To enable them in other operations, rebuild TensorFlow with the appropriate co
1/1 [=====] - 0s 73ms/step
1/1 [=====] - 0s 15ms/step
...
Se leyó adecuadamente el ángulo de arreglo de la junta 1 y es: 0.2
...
Process finished with exit code 0
```

Figura 29: Resultado de reconocimiento OCR.

# CAPÍTULO 8

---

## Análisis comparativo de sensores ópticos

---

Para la comparación de distintos sensores ópticos se realizaron dos tipos de métricas que funcionan secuencialmente. La primer métrica se basa en puntos de interés al momento de la búsqueda de compra de cámaras en el mercado y la segunda es un algoritmo de calificación de las cámaras según su habilidad de obtener características.

### 8.1. Calificador por puntos de interés

Este método utiliza las características de la cámara como organizadores, es decir, mientras mejores sean estos puntos de interés, mejor resultado tendrá la cámara cuando se es comparada con sensores ópticos. Los puntos de interés se detallan en el Cuadro 1.

Punto de interés	Cambio	Puntos de aumento	Razón del cambio
Tipo de sensor	+	1	ccd(calidad) vs cmos(velocidad)
Dimensión diagonal sensor	+	1	sensor mas grande, mas calidad
FOV	-	1	mas área menos calidad
Conectividad	+	1	mas versatilidad
Resolución	+	2	mas resolución mejores resultados
Precio	-	1	relación calidad/precio
Disponibilidad	+	2	facilidad de compra

Cuadro 1: Esta tabla es de creación propia con los puntos de interés utilizados para evaluar de las cámaras.

Los valores son dados por los fabricantes de los sensores ópticos. Se utilizan estos valores para comparar entre varios sensores y determinar el sobresaliente de esta comparación.

## 8.2. Calificador por extracción de características

Este algoritmo calificador utiliza herramientas de la librería de OpenCV para Python para obtener imágenes de prueba de la cámara. Las imágenes de prueba obtenidas se detallan a continuación:

1. Formato de prueba: Este formato contiene bordes, letras, líneas y círculos de manera que las pruebas recolecten las características adecuadamente.
2. Detectores de ejes: En esta prueba se utilizaron los detectores Canny con umbrales de valor 150, Sobel compuesto con kernel de tamaño  $3 \times 3$  y Laplaciano con kernel de tamaño  $3 \times 3$ .
3. Umbralización: Se utilizaron 7 distintos tipos de umbralización:
  - Umbralización binaria.
  - Umbralización binaria inversa.
  - Umbralización truncada a cero.
  - Umbralización truncada.
  - Umbralización truncada a cero inversa.
  - Umbralización adaptativa gaussiana.
  - Umbralización adaptativa media.
4. Transformador de Hough para líneas y círculos: se utilizaron parámetros de umbral de 150 y 200 respectivamente. Para los círculos se utilizó un umbral de tamaño mínimo de círculo de 40 píxeles para que no detectara los defectos en la hoja y únicamente detectara los círculos en negro.

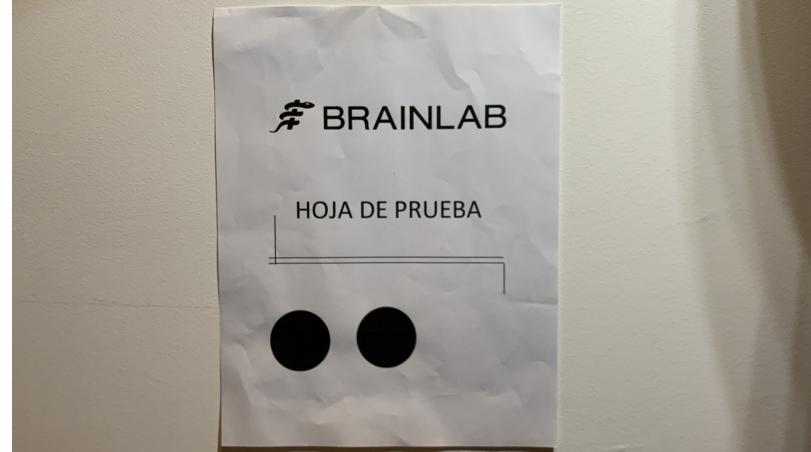


Figura 30: Formato de prueba utilizado para calificación



Figura 31: Ejemplo de imagen de prueba de detectores de ejes

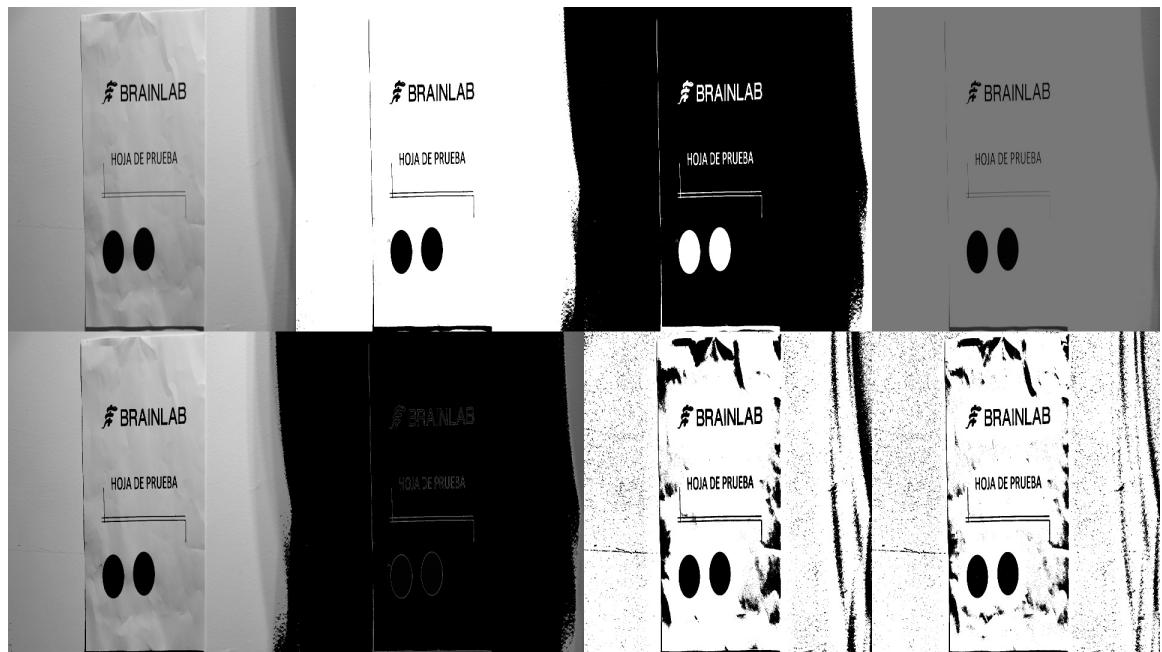


Figura 32: Ejemplo de imagen de prueba para umbralización.

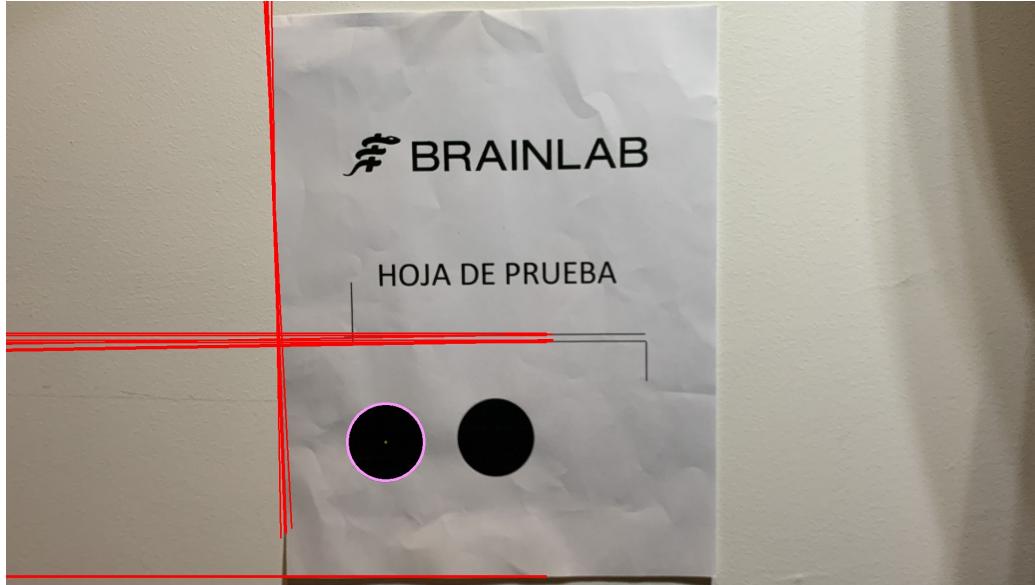


Figura 33: Ejemplo de imagen de prueba de transformador de Hough para lineas y círculos

En la Figura 31 se observan los resultados de procesamiento de imágenes en una imagen de cuadricula en la que se contienen tanto la imagen original como los distintos métodos de detección de bordes.

En la Figura 32 se observan los resultados de los siete procesamiento para la umbralización. Para su comparación se incluyo la imagen original en la cuadricula. Personalmente el resultado que mejor determina las diferencias en la imagen parece ser la umbralización binaria inversa.

En la Figura 33 se pueden ver los resultados de los transformadores de Hough sobrepuertos en la imagen original, se observan las lineas que fueron detectadas y los círculos con su centroide detectados.

Se utilizaron las pruebas de procesamiento de imágenes dado que lo que se buscó en un principio era determinar la capacidad de detección de características de cada cámara que se probó. Así mismo se introdujo la prueba de umbralización vista en la Figura 32 dado que para el programa principal de búsqueda de variables en la pantalla del laboratorio de HUMANA fue necesario implementar una serie de pre-procesamiento de la captura para tener una lectura mas clara.

No.	Dato	Información
1	CPU	Ryzen 9 3900x 3.8GHz 12 núcleos 24 hilos
2	GPU	Nvidia RTX 2080TI 12GB
3	RAM	32GB 3600Mhz
4	Sistema Op	Windows 11 x64

Cuadro 2: Datos de la computadora con la que se probó el algoritmo calificador.

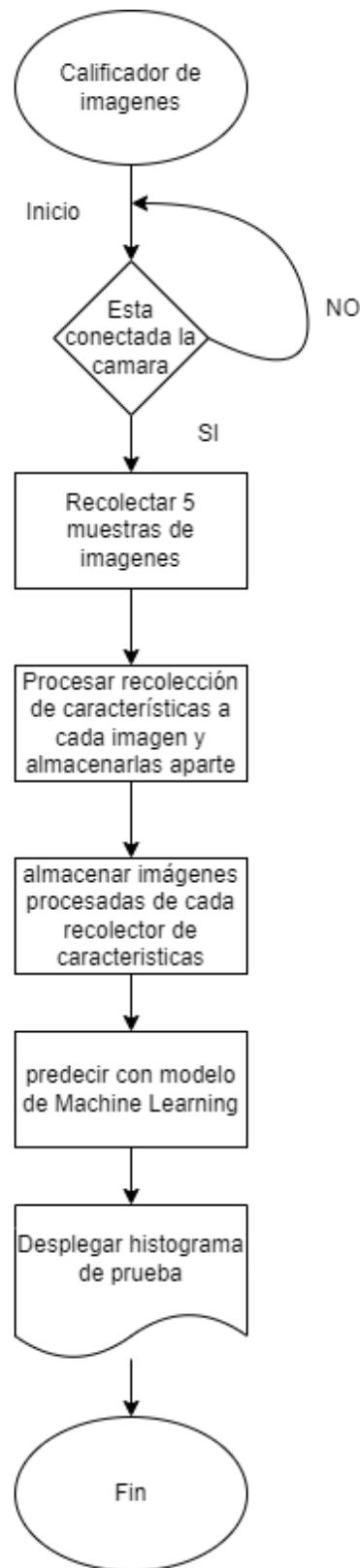


Figura 34: Diagrama de funcionalidad del algoritmo calificador de detección de características.

En la Figura 34 se describe la funcionalidad del algoritmo para realizar comparaciones de los sensores ópticos. Comenzando por una recolección de 5 imágenes que son procesadas por los tres tipos de prueba, realce de bordes, umbralización y transformadores de Hough como se observa en la Figura 31, en la Figura 32 y en la Figura 33.

Para realizar una comparación de la capacidad de lectura de características se tomaron en cuenta 3 distintas cámaras las cuales se listan a continuación:

1. Logitech C920s:

- Resolución:  $1920 \times 1080$
- Tasa de refresco:  $30 \text{ hz}$
- dFOV:  $78^\circ$

2. Logitech C270:

- Resolución:  $1280 \times 720$
- Tasa de refresco:  $30 \text{ hz}$
- dFOV:  $55^\circ$

3. Enow HD

- Resolución:  $1280 \times 720$
- Tasa de refresco:  $50 \text{ hz}$
- hFOV:  $110^\circ$

Con el calificador por puntos de interés la cámara se tuvo acceso a estas tres cámaras para realizar la prueba y según la comparación por puntos de interés la que tenía mejores especificaciones para la aplicación es la cámara Logitech c920s con una puntuación de siete puntos. La cámara Logitech c270 quedó en segundo lugar con cinco puntos y de último la cámara Enow HD con dos puntos.

Luego de realizar la calificación por puntos de interés, se realizaron las pruebas por el calificador por extracción de características. Como parámetros se utilizó una velocidad de aprendizaje de 0.01, un momento del 82 % y 5 épocas.

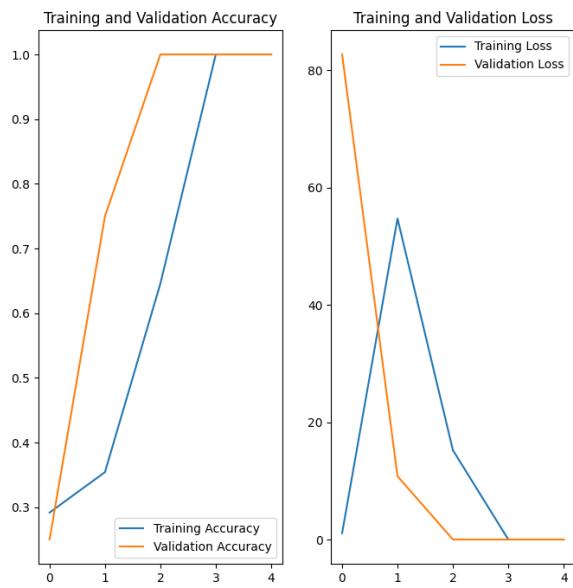


Figura 35: Resultados de entrenamiento de modelo con parámetro de 5 épocas.

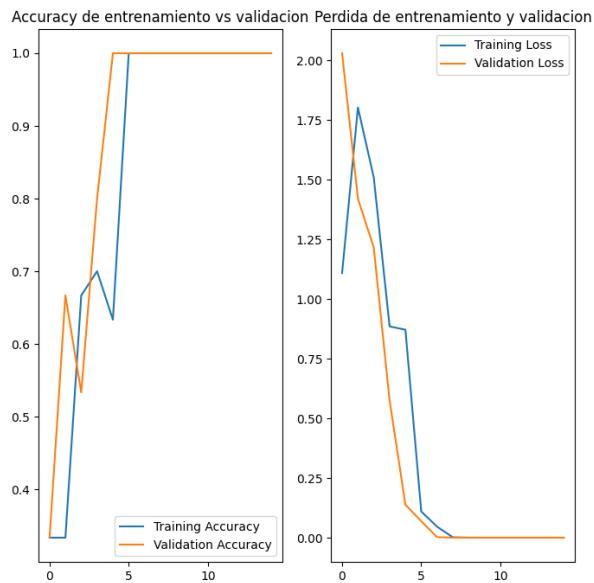


Figura 36: Resultados de entrenamiento de modelo con parámetro de 15 épocas.

En la Figura 35 se observa que el modelo no tiene sobre-ajuste y que conforme avanza el aprendizaje el modelo de validación es mas preciso. Comparando con la Figura 36 se verifica que el modelo llega a ser mas preciso pero también tiende a sobre calificar las imágenes de

prueba que se le ingresen. Para las 5 épocas se tiene un coste computacional de 5 minutos de cálculo y para 15 épocas se tiene un coste computacional de 27 minutos de cálculo. El coste computacional se determino utilizando la computadora descrita en el Cuadro 2.

```
CAMARA LOGITECH C920s
Esta cámara obtiene la calificación de 99.89 porciento en la primer prueba
1/1 [=====] - 0s 184ms/step
Esta cámara obtiene la calificación de 92.04 porciento en la segunda prueba
1/1 [=====] - 0s 182ms/step
Esta cámara obtiene la calificación de 96.81 porciento en la tercer prueba
Se le da una calificación global de 96.25 porciento a la cámara Logitech C920s
1/1 [=====] - 0s 221ms/step
```

Figura 37: Resultados del calificador con cámara c920.

```
CAMARA LOGITECH C270
Esta cámara obtiene la calificación de 96.85 porciento en la primer prueba
1/1 [=====] - 0s 194ms/step
Esta cámara obtiene la calificación de 91.50 porciento en la segunda prueba
1/1 [=====] - 0s 183ms/step
Esta cámara obtiene la calificación de 93.59 porciento en la tercer prueba
Se le da una calificación global de 93.98 porciento a la cámara Logitech C270
1/1 [=====] - 0s 186ms/step
```

Figura 38: Resultados del calificador con cámara c270.

```
CAMARA ENOW HD
Esta cámara obtiene la calificación de 92.90 porciento en la primer prueba
1/1 [=====] - 0s 235ms/step
Esta cámara obtiene la calificación de 85.60 porciento en la segunda prueba
1/1 [=====] - 0s 200ms/step
Esta cámara obtiene la calificación de 90.04 porciento en la tercer prueba
Se le da una calificación global de 89.51 porciento a la cámara Enow HD
```

Figura 39: Resultados del calificador con cámara Enow.

En la Figura 37 se visualiza los resultados obtenidos por el calificador de la cámara Logitech C920s con un 96.25 %. En la Figura 38 se visualizan los resultados obtenidos por el calificador de la cámara Logitech c270 con un 93.98 %. En la Figura 39 se visualizan los resultados obtenidos por el calificador de la cámara Enow con un 89.51 %.

La mejor cámara para reconocimiento de caracteres fue la cámara Logitech C920s según ambas métricas de calificación. Las mejores capacidades de resolución y el sensor mas grande de la cámara Logitech C920s permiten que el software reconozca mejor las características que fueron probadas y que fueron útiles para el sistema principal de reconocimiento de variables de Varioguide.

## CAPÍTULO 9

---

### Envío de datos

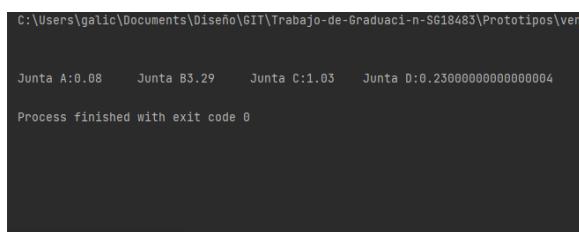
---

Luego del procesamiento de imágenes utilizando Tesseract (en el lenguaje Python) se requiere enviar mediante un método estandarizado los datos de calibración obtenidos de la pantalla del laboratorio de HUMANA. El propósito de enviar datos de calibración es para automatizar mecánicamente los procesos de calibración quirúrgica en HUMANA haciendo uso de microcontroladores.

Para este envío de datos se consideraron 2 métodos de envío de datos que soportasen la mayoría de microcontroladores modernos. Los métodos considerados son:

- Envío de datos por serial USB.
- Envío de datos por TCP.

Para el envío de datos se necesita manipular los datos ya que el algoritmo de reconocimiento de variables obtiene los datos de calibración en formato “String” de Python. Se realiza un proceso de conversión a paquetes de datos en bytes utilizando codificación UTF-8. Se envía cada byte secuencialmente al microcontrolador “esclavo” en cuanto este disponible la conversación serial.



A screenshot of a terminal window showing the command-line interface. The window title is 'C:\Users\galic\Documents\Diseño\GIT\Trabajo-de-Graduaci-n-SG18483\Prototipos\venv'. The text displayed in the terminal is:  
Junta A:0.08      Junta B3.29      Junta C:1.03      Junta D:0.23000000000000004  
Process finished with exit code 0

Figura 40: Prueba de envío de variables de calibración en formato serial.

Para el envío TCP se utilizo la librería de funciones “Socket” de Python en el que se utilizaba un ip en la red y un puerto al que se le enviaron los datos. El programa utiliza la computador como un servidor y el micro controlador ESP-32 como cliente TCP para recibir los datos. Ambos servidor y cliente están conectados a una red en la por un puerto mandan datos. Para este tipo de envío de datos se utilizaron datos en formato “Float” ya que de esta manera el mensaje se logró enviar toda la información de una junta al mismo tiempo empaquetada en un JSON utilizando codificación ASCII. Para su codificación se utilizo un mensaje que incluye el numero de Junta, el signo y el ajuste de la junta.

```

COM4
Connecting to WiFi..
Connected to the WiFi network
192.168.5.158
Connection Sucessful.
Receiving...
1.1
0.2
0.0

Autoscroll Mostrar marca temporal Nueva línea 115200 baudio Limpiar salida

```

Figura 41: Captura de pantalla de prueba de envío de variables de calibración en formato TCP.

En la Figura No.41 se observa una prueba satisfactoria de envío de datos por medio de TCP. Se enviaron los datos de la siguiente manera:

- Junta: Esta se envía “1.1” si es la Junta 1, “2.2” si es la Junta 2 y “3.3” si es la Junta 3.
- Ajuste: Este se llena con el ajuste desplegado de el sistema OCR.
- Signo: “0.0” si es positivo el ajuste y “1.1” si es negativo el ajuste.
- Ángulo o desplazamiento: Si es la junta 3 se agrega este último diciendo “1.0” para mm o “0.0” para °.

# CAPÍTULO 10

---

## Anclaje físico de cámara

---

Para el anclaje de la cámara utilizado, se definió el problema y los parámetros con los que se trabajó. Así mismo se realizo un diseño CAD para una primer iteración el cual se fue modificando según se realizaron pruebas utilizando impresión 3D.

### 10.1. Definición del problema

El sistema de calibración VarioGuide de Brainlab en HUMANA despliega los valores de calibración en una pantalla montada en un brazo mecánico articulado . Esta pantalla es un modelo patentado por Brainlab por lo que no presenta las cualidades de un tipo anclaje estándar como por ejemplo el estándar “VESA 100”. Se puede ver con más detalle el laboratorio de HUMANA en la Figura 42. Dada la dificultad para acceder a el laboratorio de HUMANA se utilizó como referencia para pruebas un monitor marca Dell con modelo e2220H con las funcionalidades presentadas en el Cuadro 3.

Dimensión	Medida	Dimensionales
Alto	32.21	cm
Ancho	53.78	cm
Diagonal	23.8	pulgadas

Cuadro 3: Dimensiones del monitor utilizados en los experimentos.

El monitor Dell presenta las siguientes características técnicas:

- Resolución  $1920 \times 1080$
- Brillo de  $250 \frac{cd}{m^2}$
- 92 PPI
- Aspecto 16:9
- Panel tipo W-LED “IPS”

Para realizar un sistema de anclaje mecánico adecuado del sensor óptico a utilizar se tomaron en cuenta las siguientes consideraciones:

- Debe de poder anclarse utilizando soportes propios.
- Debe de poder soportar al menos 1 Kg en carga estática para soportar el peso de la cámara.
- Debe de poder cambiar la posición de la cámara para que la pantalla quede sin obstáculos enfrente.
- Debe de ser modular.
- Debe de utilizarse un software para su diseño.
- Debe de utilizar materiales inoxidables (no contaminantes).



Figura 42: Pantalla del laboratorio de HUMANA.

## 10.2. Primer Prototipo

Para la primer iteración se utilizó el software de Autodesk Inventor Profesional 2023 para realizar el diseño tridimensional de el brazo rígido a soportar el sensor óptico. También se utilizó este software para simular el mecanismo acoplado a la pantalla. Realmente es conveniente trabajar esta simulación en un ambiente tipo “ensamblaje” para poder tomar en cuenta las dimensiones del monitor. En este prototipo crea un importante enlace entre el diseño en software CAD y procesos de manufactura como lo es la impresión 3D y la compra de materiales externos. En este enlace se consideran las restricciones de tamaño de límites máximos y mínimos tanto del monitor como de la extrusora 3D.

Para esta primer iteración se utilizó la aproximación de soportes por prensa en “c”, ya que esta aproximación permitiría que con unos simples ajustes al diseño se pueda implementar al laboratorio de HUMANA. En la Figura 43 se puede ver el frente del monitor Dell con el acople ensamblado. y en la Figura 44 se puede ver el acople por la parte trasera del monitor. Para el diseño CAD se utilizaron barras roscadas inoxidables métricas M5 y tubos galvanizados de  $\frac{1}{2}$  pulgada. Dado que se requiere poder mover la cámara de enfrente de la pantalla, se utilizó la bisagra vista en la Figura 45.

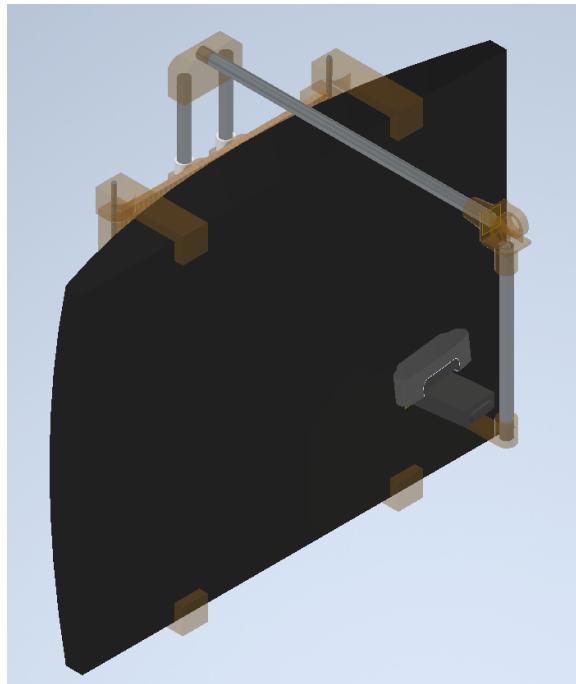


Figura 43: Ensamblaje CAD de primer prototipo de anclaje de cámara por enfrente.

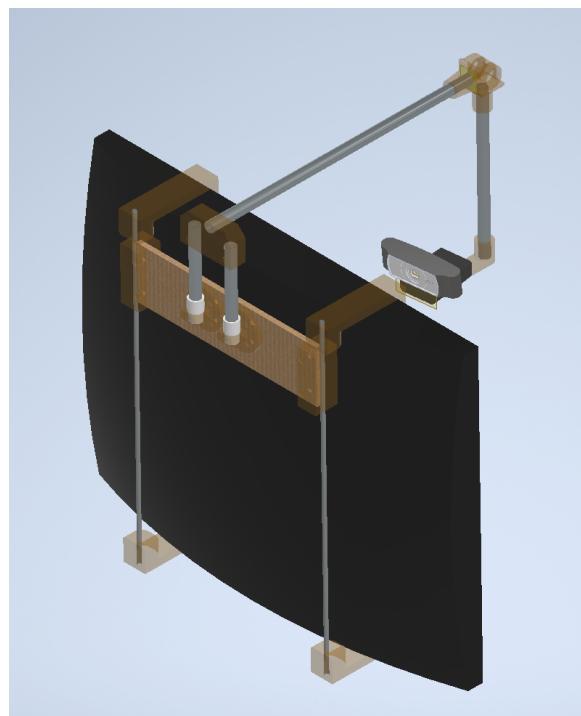


Figura 44: Ensamblaje CAD de primer prototipo de anclaje de cámara por atrás.

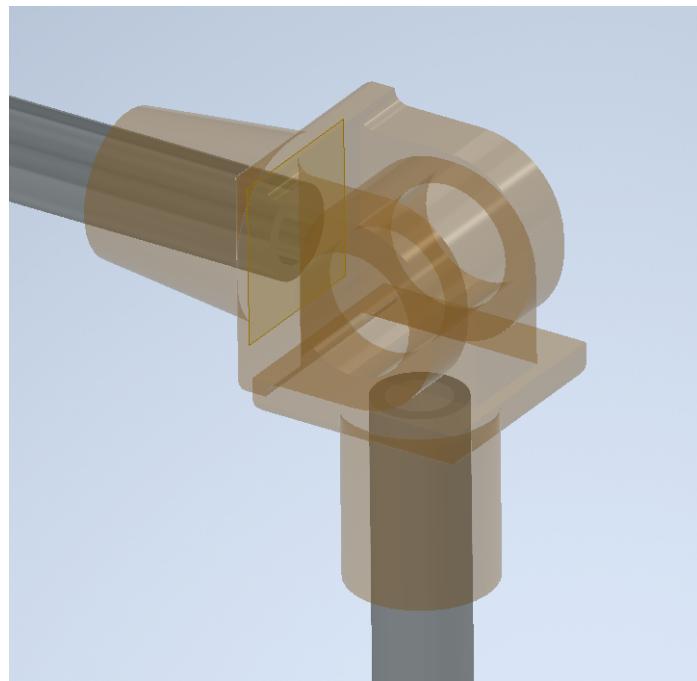


Figura 45: Bisagra en diseño de software CAD.

Para determinar la distancia de la cámara a la pantalla se trabajó con la ecuación 1 y se trabajo con la cámara Logitech C920s dando como resultado una distancia de 27.52 cm para visualizar la totalidad de los 40 cm de altura del monitor.



Figura 46: Resultado de prensa C fabricada.

En la Figura 46 se visualiza el primer problema que existió sobre la prensa C que se utiliza como método de acoplamiento al monitor. Este método necesitaba una forma de fijarse a la geometría del monitor por la parte de atrás. Para solucionar este problema se realizó una nueva iteración de sistema de acople.

### 10.3. Segundo prototipo

Para el segundo prototipo se realizó un nuevo diseño del monitor al que el sistema se acopló. Este diseño fue más detallado y se utilizaron las dimensiones del fabricante Dell para poder realizarse este diseño en el software Autodesk Inventor Professional 2023. Se mejoró este diseño desde el anterior dado que de esta manera se pudo disminuir los errores al replicar el diseño en pruebas físicas.



Figura 47: Ensamblaje de prototipo de acople No.2 en Inventor.



Figura 48: Ensamblaje de prototipo de acople No.2 en Inventor posterior.

En esta segunda iteración del sistema de acople se utilizó un soporte en su totalidad por los marcos superiores del monitor. Se utilizó corte láser de un material MDF e impresiones 3D en material PLA para llevar a cabo su fabricación después de ser aprobado el diseño en CAD. En la Figura 47 se visualiza el segundo prototipo en un ensamblaje en Autodesk Inventor Professional 2023.

Para realizar los cálculos de esfuerzos en la estructura mecánica se recurrió a la Ecuación

6 y a la Ecuación 8 en las que se tuvieron los siguientes parámetros:

- Carga V =  $1Kg \times g = 9.81N$
- Área A =  $2cm^2$
- Segundo momento de área I =  $\frac{a \times b^3}{12} = 1.66 \times 10^{-9}m^4$
- Distancia de carga hacia soporte y = 0.3m

Se determino un esfuerzo cortante por flexión máximo de 73,576 (Pa). El esfuerzo admisible cortante por flexión es de 33,000,000 (Pa). Se obtuvo un esfuerzo normal por flexión de 141.26 (MPa). El esfuerzo normal por flexión admisible es de 2,300 (MPa). Según estos resultados se tuvo un factor de seguridad de al menos  $n = 16$ .

Al momento de realizar las pruebas se encontró que el modelo soportaba adecuadamente el peso de la cámara Logitech C920s dado que el mecanismo esta diseñado para soportar una cámara de hasta 16 Kg. En este prototipo se determinó que el punto critico de sobrecarga de estrés de cargas se encuentra en las esquinas en las que se soporta del monitor.

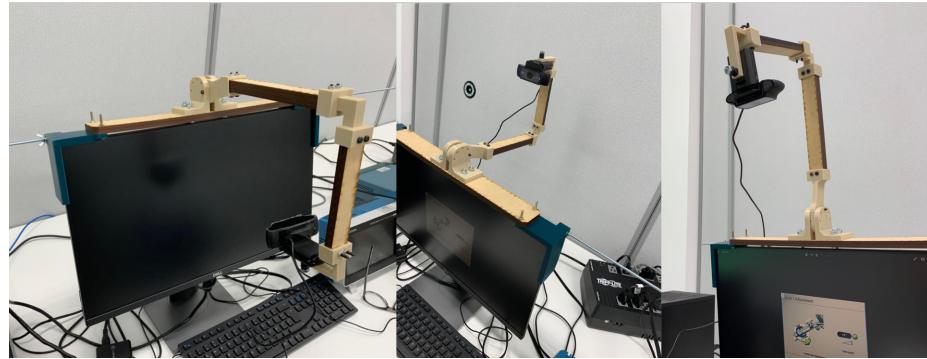


Figura 49: Ensamblaje de prototipo de acople No.2.

Dado que se tenia en cuenta que el modelo debía incluir materiales que no fueran a corroerse por las restricciones quirúrgicas. Se investigó una pintura para aplicarse a la totalidad del prototipo. Se encontró la pintura “High Standard 99.9 % Antibacterial”. Se consideró esta pintura como recubrimiento dado que cumple con las normas ISO 221196:2007 de prueba de actividad antibacterial sobre superficies plásticas y otras superficies. Esta pintura es un recubrimiento acrílico para ambientes de desinfección rutinaria. Se consideró como la mejor opción para recubrir el sistema de acople al momento de ser implementada en el laboratorio de HUMANA, ya que aunque no se desinfecte con tanta regularidad, cuenta con la norma ASTM D5590-00 que indica una resistencia al ataque de hongos durante al menos 2 años.

# CAPÍTULO 11

---

## Conclusiones

---

La mejor cámara disponible para el sistema OCR fue la cámara Logitech C920s según los resultados analíticos presentados.

El sistema de envío de datos funciona adecuadamente enviando los datos importantes de ajuste de junta mecánica byte por byte en metodo Serial.

Utilizando formato JSON el programa es capaz de enviar datos en método TCP de una computadora a un microcontrolador ESP32.

Un sensor óptico con gran angular no es de beneficio para la aplicación en cuestión. En los resultados se tienen dos sensores con la misma resolución pero con un mayor grado de FOV. El de mayor FOV (Enow HD) resultó con una calificación mas baja que el de menor dFOV (Logitech C270).

El sistema de acople resultó lo suficientemente estable para ser implementado en la pantalla del laboratorio de HUMANA soportando al menos 16 Kg antes de la fractura.

El sistema de reconocimiento de variables utiliza una red neuronal de aprendizaje automático con complejidad computacional suficiente como para estresar un procesador de bajo consumo.

El modelo de la red neuronal para detección de dígitos no presenta sobre ajuste, ni sub ajuste. También presenta una precisión del 96 % al reconocer y enlazar el dígito a su set de datos MNIST.

El modelo de la red neuronal no presenta sobre carga al procesador ya que es un modelo que se almacenó y se puede cargar cada vez que el programa lo necesite para realizar detección OCR.

El archivo del modelo no supera los 16 MB por lo que no presenta una amenaza al almacenamiento al momento de la implementación en los laboratorios de HUMANA.

# CAPÍTULO 12

---

## Recomendaciones

---

En futuras interacciones de reconocimiento de variables se recomienda utilizar ángulo de visión reducido para la cámara ya que la distorsión óptica aumenta mediante se incrementa el ángulo de visión.

Para futuras iteraciones se recomienda hacer mas liviano el sistema de acople mediante un análisis de esfuerzos críticos.

Dada la complejidad computacional para reconocimiento de variables, se recomienda utilizar una cámara con mejor resolución y aumentar la capacidad computacional en HUMANA.

Se recomienda al momento de implementación del acople mecánico utilizar un recubrimiento como el propuesto en el capítulo 10 ya que de esta manera se podría continuar el ambiente esterilizado de alta importancia en los laboratorios de HUMANA.

Para futuras iteraciones se recomienda tener una banco de fotografías actualizadas del software de calibración de VarioGuide de Brainlab, preferiblemente del laboratorio de HUMANA de esta manera se pueden realizar mas pruebas con mas combinaciones de dígitos y optimizar mejor el sistema.

Para futuras iteraciones se recomienda reducir el sistema de reconocimiento de variables en un método embebido o compactos como un RaspberryPi o una cámara JeVois para liberar espacio de trabajo útil en el laboratorio de HUMANA.

## CAPÍTULO 13

---

### Bibliografía

---

- [1] Renishaw plc, *Sistema robotizado neuromate® para neurocirugía estereotáctica* . Accessed Apr. 19, 2022 [Online]. dirección: <https://www.renishaw.es/es/sistema-robotizado-neuromate-para-neurocirugia-estereotactica--10712>.
- [2] *Información Acerca de Brainlab*, Accessed: 2022-05-03. dirección: <https://www.brainlab.com/es/acerca-de-brainlab/>.
- [3] J. González, «Reconocimiento de objetos utilizando OpenCV y Python en una Raspberry pi 2 en una tlapalería.,» Tesis de licenciatura, Universidad Autónoma del Estado de México, 2017. dirección: <http://ri.uaemex.mx/bitstream/handle/20.500.11799/68150/tesis%5C%20opencv%5C%20python.pdf?sequence=1>.
- [4] G. Viera-Maza, «Procesamiento de Imágenes usando OpenCV aplicado en Raspberry PI para la clasificación del cacao.,» Tesis de licenciatura, Universidad de Piura, 2017. dirección: [https://pirhua.udep.edu.pe/bitstream/handle/11042/2916/IME\\_218.pdf](https://pirhua.udep.edu.pe/bitstream/handle/11042/2916/IME_218.pdf).
- [5] M. E. González, «Técnicas de procesamiento de imágenes aplicadas al monitoreo de procesos alimentarios.,» Tesis de mtría., Universidad de la República de Uruguay, 2021. dirección: <https://www.colibri.udelar.edu.uy/jspui/bitstream/20.500.12008/26835/1/GON21.pdf>.
- [6] A. Oliva, «Desarrollo de una aplicación de reconocimiento de imágenes utilizando Deep Learning con OpenCV.,» Tesis de licenciatura, Universidad Politécnica de Valencia, 2018. dirección: <https://riunet.upv.es/bitstream/handle/10251/107243/OLIVA%5C%20-%5C%20Desarrollo%5C%20de%5C%20una%5C%20aplicaci%5C%C3%5C%B3n%5C%20de%5C%20reconocimiento%5C%20en%5C%20im%5C%C3%5C%A1genes%5C%20utilizando%5C%20Deep%5C%20Learning%5C%20con%5C%200...pdf?sequence=1&isAllowed=y>.
- [7] M. Forero y J. Henao, «Sistema de control de calidad por procesamiento de imágenes.,» Tesis de licenciatura, Pontificia Universidad Javeriana, 2016. dirección: <https://repository.javeriana.edu.co/bitstream/handle/10554/21451/ParedesForeroMelissa2016.pdf?sequence=1&isAllowed=y>.

- [8] OMS, «Más de la mitad de las personas con epilepsia no reciben ningún tipo de atención en América Latina y el Caribe,» *OPS/OMS Guatemala*, 2019. dirección: [https://www3.paho.org/gut/index.php?option=com\\_content&view=article&id=1188:mas-de-la-mitad-de-las-personas-con-epilepsia-no-reciben-ningun-tipo-de-atencion-en-america-latina-y-el-caribe&Itemid=441](https://www3.paho.org/gut/index.php?option=com_content&view=article&id=1188:mas-de-la-mitad-de-las-personas-con-epilepsia-no-reciben-ningun-tipo-de-atencion-en-america-latina-y-el-caribe&Itemid=441).
- [9] *Humana*, Accessed: 2022-04-29. dirección: <https://humanagt.org/>.
- [10] J. P. Guerra, «Algoritmos de Visión por Computadora para el Reconocimiento de la Pose de Agentes Empleando Programación Orientada a Objetos y Multihilos.,» Tesis de licenciatura, Universidad Del Valle de Guatemala, 2021.
- [11] J. I. Ramírez, «Herramienta de Software de Visión por Computadora para Aplicaciones de Robótica de Enjambre en una Mesa de Prueba - Fase III.,» Tesis de licenciatura, Universidad Del Valle de Guatemala, 2021.
- [12] J. G. González, «Diseño de un sistema de procesamiento de imágenes para la identificación de variables en la pantalla del sistema Brainlab del Centro de Epilepsia y Neurocirugía Funcional HUMANA.,» Tesis de licenciatura, Universidad Del Valle de Guatemala, 2021.
- [13] Brainlab AG, *High precision instrument alignment VarioGuide*, Accessed Apr. 27, 2022 [Online]. dirección: <https://www.brainlab.com/surgery-products/overview-neurosurgery-products/cranial-navigation/>.
- [14] A. Ajgaonkar, «The Value of Computer Vision: More Than Meets the Eye,» *Tech Journal*, n.º 1, mar. de 2021.
- [15] N. K. Mohamed Cheriet y col., *Character Recognition Systems [OCR]*. Wiley-Interscience, 2007, ISBN: 9780471415701. dirección: <http://gen.lib.rus.ec/book/index.php?md5=dc99595b218c296c424678ae17925fa8>.
- [16] Y. Fernández, «Live Text en iOS 15: cómo usarlo para reconocer textos con la cámara o desde tu galería de fotos,» *Xataka*, sep. de 2021.
- [17] BBC, «How do digital images work?» *BBC Bitesize*, mar. de 2022. dirección: [www.bbc.co.uk/bitesize/topics/zf2f9j6/articles/z2tgr82](https://www.bbc.co.uk/bitesize/topics/zf2f9j6/articles/z2tgr82).
- [18] M. Lyra, A. Ploussi y A. Georgantzoglou, «MATLAB as a Tool in Nuclear Medicine Image Processing,» en oct. de 2011, ISBN: 978-953-307-907-3. DOI: 10.5772/19999.
- [19] S. Gomez, «Qué es y como mejorar la nitidez de una imagen,» oct. de 2027. dirección: <https://miniminim.com/nitidez-de-una-imagen/>.
- [20] J. Barros, «Fotografía: tamaño de imagen, resolución y tamaño físico,» abr. de 2022. dirección: <https://perio.unlp.edu.ar/catedras/iddi/wp-content/uploads/sites/125/2022/04/Fotografia.-Tamano-de-Imagen-resolucion-y-tamano-fisico.pdf>.
- [21] Sanne, «What is the difference between optical zoom and digital zoom?» *CoolBlue*, ene. de 2022. dirección: <https://www.coolblue.nl/en/advice/difference-optical-and-digital-zoom.html>.
- [22] Y. Ananenko, «La cámara fotográfica y la cámara móvil.Comparación de características y prestaciones.Análisis de parámetros de calidad.,» Tesis de mtría., Universitat Politècnica de Catalunya, jun. de 2016. dirección: [https://upcommons.upc.edu/bitstream/handle/2117/118954/yulia.ananenko%20-%20TFG\\_Yulia%20Ananenko.pdf](https://upcommons.upc.edu/bitstream/handle/2117/118954/yulia.ananenko%20-%20TFG_Yulia%20Ananenko.pdf).

- [23] D. Berkenfeld, M. Corrado y L. Silverman, «Entendiendo la Distancia Focal,» *Nikon*, mar. de 2020. dirección: <https://www.nikon.com.mx/learn-and-explore/a/tips-and-techniques/entendiendo-la-distancia-focal.html#>.
- [24] D. Zafra, «Qué es la distancia focal y el campo de visión en fotografía,» *Capture the atlas*, ene. de 2022. dirección: <https://capturetheatlas.com/es/que-es-la-distancia-focal/>.
- [25] N. León, «ISO en Fotografía: Qué es y Cómo Se Usa,» *dzom*, ene. de 2021. dirección: <https://www.dzoom.org.es/sensibilidad-iso-que-es-y-como-funciona/>.
- [26] N. Mansurov, «What is ISO? The Complete Guide for Beginners,» *Photography life*, abr. de 2022. dirección: <https://photographylife.com/what-is-iso-in-photography>.
- [27] A. K. Roy, *Principles of colour and appearance measurement: Object appearance, colour perception and instrumental measurement*, ép. Woodhead Publishing Series in Textiles 159. Woodhead Publishing, 2014, vol. Volume 1, ISBN: 978-0-85709-229-8.
- [28] S. McHugh, «DIGITAL CAMERA SENSORS,» *Cambridge in Colour*, mayo de 2020. dirección: <https://www.cambridgeincolour.com/tutorials/camera-sensors.htm>.
- [29] J. P. David A. Forsyth, *Computer Vision: A Modern Approach*, US ed. Prentice Hall, 2002, ISBN: 9780130851987. dirección: <http://gen.lib.rus.ec/book/index.php?md5=9e052a7bd5eb18ca729f350b4c629a89>.
- [30] J. Russ, *The Image Processing Handbook*, 4th. CRC Press, 2002, ISBN: 0-8493-1142-X.
- [31] B. Medina, «Convolución en 2D (Filtrado espacial),» *Procesamiento digital de imágenes e Inteligencia Artificial GitHub*, jun. de 2019. dirección: <https://bryanmed.github.io/conv2d/>.
- [32] M. G. Aditi Majumder, *Introduction to Visual Computing: Core Concepts in Computer Vision, Graphics, and Image Processing*, 1.<sup>a</sup> ed. CRC Press, 2018, ISBN: 978-1-4822-4491-5.
- [33] R. Smith, «Tesseract,» *Github*, jul. de 2022. dirección: <https://github.com/tesseract-ocr/tesseract>.
- [34] A. C. Ian Goodfellow Yoshua Bengio, *Deep Learning [pre-pub version]*. MIT Press, 2016. dirección: <http://gen.lib.rus.ec/book/index.php?md5=a05be4942325aea362e2aff8c305b0>
- [35] L. M. John Paul Mueller, *Machine Learning For Dummies*, 1.<sup>a</sup> ed., ép. Computer/Tech. For Dummies, 2016, ISBN: 9781119245513. dirección: <http://gen.lib.rus.ec/book/index.php?md5=3106fc3e11b5babf818f7441430c2776>.
- [36] JavaScript, *Introducing JSON*, <https://www.json.org/json-en.html>, 1999.
- [37] K. N. Richard Budynas, *Shigley's Mechanical Engineering Design, 9th Edition*, 9.<sup>a</sup> ed., ép. McGraw-Hill Series in Mechanical Engineering. McGraw-Hill, 2010, ISBN: 9780073529288. dirección: <http://gen.lib.rus.ec/book/index.php?md5=3fc7c88cb2a5f16a20ef69905c028d99>.
- [38] C. Liao, *OCR Engine Comparison - Tesseract vs. EasyOCR*, <https://medium.com/swlh/ocr-engine-comparison-tesseract-vs-easyocr-729be893d3ae>, 2020.

# CAPÍTULO 14

---

## Anexos

---

### 14.1. GitHub

Para documentar el desarrollo del proyecto se utilizó la plataforma de GitHub haciendo un historial de los avances realizados. Este contiene los archivos y guías de usuario necesarias para replicar este proyecto.

La dirección del repositorio se puede encontrar en el siguiente enlace: <https://github.com/SSGR18483/Optimizacion-del-reconocimiento-de-variables-de-Varioguide.git>

