
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.

Juan Gerardo González Morales



UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



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.

Trabajo de graduación presentado por Juan Gerardo González Morales para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2022

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



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.

Trabajo de graduación presentado por Juan Gerardo González Morales para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2022

Vo.Bo.:

(f) 
Ing. Kurt Kellner

Tribunal Examinador:

(f) 
Ing. Kurt Kellner

(f) 
MSc. Carlos Esquit

(f) 
Dr. Juan Carlos Lara

Fecha de aprobación: Guatemala, 7 de Enero de 2022.

La robótica siempre ha sido una de mis metas desde que tengo conocimiento de la existencia de estas máquinas, me gustaba bastante ver como se hacen los robots en programas de televisión y siempre tuve el deseo de hacer un proyecto con mis compañeros para poder desarrollar algo cuando tuviera la oportunidad. Por otra parte, es bastante amplia las oportunidades que se pueden tener si aprovecho esta rama de la ciencia para poder ayudar a las personas y ayudar al país.

La programación también es de bastante interés para mí, siempre he querido poder ayudar a desarrollar aplicaciones, videojuegos y robots, etc. Con los conocimientos adquiridos en la programación a lo largo de la carrera, culminan en esta área y los pude aplicar para poder automatizar el proceso de un brazo asistencia de operaciones cerebrales.

Agradezco también a mis hermanos, quienes estuvieron siempre conmigo y me ayudaron a cumplir mis metas. Quiero agradecerle a mis compañeros del campus por los buenos momentos que pasamos juntos y por ayudarme a presentar este proyecto. También le agradezco especialmente a mi papá y mamá que siempre estuvieron ayudándome con lo que podían en las situaciones más difíciles, y quiero darle un agradecimiento especial a mi abuela que en paz descanse que me dio ánimos para estudiar en la universidad del Valle de Guatemala y me ayudo a obtener la beca con la cual pude seguir estudiando, siempre estaré agradecido con ella.

Prefacio	III
Lista de figuras	VIII
Resumen	IX
Abstract	X
1. Introducción	1
2. Antecedentes	2
3. Justificación	5
4. Objetivos	7
4.1. Objetivo general	7
4.2. Objetivos específicos	7
5. Alcance	8
6. Marco teórico	10
6.1. La epilepsia	10
6.2. Visión por computadora	11
6.3. Python	11
6.4. Reconocimiento óptico de caracteres (OCR por sus siglas en inglés)	11
6.5. Tesseract	12
6.6. Matlab	12
6.7. BrainLab	12
6.8. Pixel	13
6.9. Cámaras digitales	13
6.10. Dispositivo de carga acoplada (CCD por sus siglas en inglés)	13
6.11. Mosaico de Bayer	14
6.12. Rojo Verde Azul (RGB por sus siglas en inglés)	15
6.13. Zoom digital	15

6.14. Zoom óptico	16
6.15. Lentes lineales	17
6.16. Sensibilidad ISO	17
6.17. Temperatura de color	18
6.18. Sharpness	19
6.19. Acutancia	19
6.20. Resolución	20
7. Sistema Varioguide de Brainlab	21
7.1. Nuevo prototipo	22
7.1.1. Interfaz principal	23
7.1.2. Clases del sistema	24
7.1.3. Interfaz automática	25
8. Sistema de visión por computadora	26
8.1. Determinación de junta a manipular	28
8.2. Filtrado de color	30
8.3. Encontrando el área de interés	31
8.4. Filtrado de área de interés	34
8.4.1. Obtención de datos por inversión de colores	36
8.4.2. Obtención de datos por binarización promediada	36
8.5. Resultados de filtrados con diferentes cámaras	40
8.5.1. Resultados con GoPro Hero 8	40
8.5.2. Camara logitech 1080p HD	44
8.6. Filtración de datos y envío de datos por serial	48
9. Implementación física del anclaje de cámara	51
9.1. Primer prototipo	51
9.2. Segundo prototipo	53
9.3. Sistema giratorio	53
9.3.1. Simulación de cargas puntuales	56
10. Conclusiones	58
11. Recomendaciones	59
12. Bibliografía	60
13. Anexos	63
13.1. Programación	63
13.2. Protipo sujetador de web cam en HUMANA	63

Lista de figuras

1.	Rosa	2
2.	Operación cerebral	6
3.	Cavidades de luz	14
4.	Filtro Bayer	15
5.	Zoom digital	16
6.	Zoom óptico	17
7.	Ajuste de balance blanco.	18
8.	Temperatura del color	18
9.	Acutancia	19
10.	Resolución	20
11.	Diseño de brazo mecánico	21
12.	Pantallas de ajustes en Brainlab	22
13.	Diagrama de flujo del programa capaz de obtener datos de las pantallas de Brainlab	23
14.	Interfaz automática	23
15.	Interfaz principal	24
16.	Aplicación del sistema principal	24
17.	Métodos de interfaz automática	25
18.	Búsqueda de área de interés obtenida mediante el sistema principal	25
19.	Sistema de reconocimiento óptico de imágenes	26
20.	Reconocimiento óptico de imágenes	27
21.	Obtención de captura de imagen	28
22.	Imagen obtenida de Brainlab	28
23.	Imagen escala a color de 24 bits	29
24.	Imagen escala a gris de 8 bits	29
25.	Buscar variables con OCR de Tesseract	30
26.	Encontró el string correcto en la posición 632	30
27.	Uso de filtro threshold	31
28.	Imagen con filtro threshold	31
29.	Filtro canny	32

30.	Matrix gaussiana	32
31.	Enfoque de pixeles	33
32.	Contorno de interés con canny	34
33.	Contorno de interés con imagen original	34
34.	Contorno de interés con resize	34
35.	Contorno en escala de gris con resize	35
36.	Contorno de interés en threshold y resize	35
37.	Valor del error de pantalla en Brainlab	36
38.	Ecuación umbral	36
39.	Threshold dinámico	36
40.	Mapa de bordes	37
41.	Filtrado de borde	37
42.	Filtrado de caracteres	38
43.	Intensidad de primer plano	38
44.	Caracter encontrado	38
45.	Periferias de las esquinas	39
46.	Intensidad media	39
47.	Binarización	39
48.	Resultado de la binarización	40
49.	Cámara GoPro	40
50.	Imagen con filtro inicial	41
51.	Imagen con filtro grises	42
52.	Imagen con filtro Threshold	42
53.	Imagen con filtro canny	43
54.	Imagen con área obtenida	43
55.	Imagen con área agrandada	43
56.	Imagen con área agrandada en Threshold	44
57.	Imagen con área con limpieza y binarización invertida	44
58.	Imagen con filtro inicial	45
59.	Imagen con filtro grises	45
60.	Imagen con filtro Threshold	46
61.	Imagen con filtro canny	46
62.	Imagen con área obtenida	47
63.	Imagen con área agrandada	47
64.	Imagen con área agrandada en Threshold	47
65.	Imagen con área con limpieza y binarización invertida	48
66.	Valor del error de pantalla en Brainlab	48
67.	Datos que espera el sistema de control principal	49
68.	Resultado de verificación de datos	50
69.	Cilindros de PLA	52
70.	Sistema giratorio	52
71.	Prototipo inicial	52
72.	Sistema giratorio	53
73.	Cilindros de acero galvanizado	53
74.	Sistema giratorio con pines	54
75.	Pieza para hacer girar la cámara	54
76.	Agarrador de cámara web simulación con GoPro	55

77.	Agarrador de cámara web simulación con Logitech	56
78.	Brazo de cámara web con Logitech	56
79.	Esfuerzos de Von Mises	57
80.	Desplazamiento del material	57
81.	Brazo de HUMANA	64

Este trabajo presenta la integración de los resultados, investigación y la metodología para poder diseñar e implementar un sistema con capacidad de reconocimiento óptico de caracteres (OCR por sus siglas en inglés) enfocado principalmente en poder obtener los datos proporcionados por el sistema de Varioguide de Brainlab y enviarlos a un sistema de control principal para automatizar el sistema mecánico con la finalidad de optimizar el tiempo entre cirugías.

El proyecto completo consta de 4 partes distribuida a 4 estudiantes. La primera parte se enfoca en el diseño de la estructura mecánica, otro en el ajuste de los motores steppers, uno en el mando de control y el último en el diseño del sistema de reconocimiento óptico de caracteres. El enfoque de este trabajo de graduación es el de diseñar un sistema de reconocimiento óptico de caracteres (OCR por sus siglas de inglés). Se diseñó un software fácil de utilizar y amigable con el usuario para facilitar el uso de este, así mismo para el análisis de la pantalla de Varioguide de Brainlab se usó una cámara web debido a su precio y definición de esta para luego hacer el procesamiento de datos de las imágenes obtenidas, y así encontrar las variables de interés que en este caso son los porcentajes de error que muestra la pantalla de Brainlab.

El sistema de Varioguide se ajusta manualmente por etapas, el cual tiene una junta esférica para ajustes gruesos y 4 juntas (3 angulares y 1 lineal) para ajustes finos. La automatización se realizó en las últimas 4 juntas del sistema ya que son las juntas que el sistema de Brainlab muestra sus porcentajes de error. Cuando el operador se encuentra en esta fase el software detecta automáticamente en cuál de los 4 motores se encuentra en el momento para luego obtener una foto, y con esta misma se procede a hacer procesamiento de imágenes con cambios matriciales para poder obtener el porcentaje de error del motor.

Al obtener el valor del porcentaje de error de un motor se hace un filtrado especial de las variables para poder verificar la información. Ya al verificar si los datos son correctos se procede a enviar los datos según indique el control principal.

This work presents the integration of the results, research and the methodology to be able to design and implement a system with character recognition capacity (OCR) mainly focused on being able to obtain the data provided by Brainlab's Varioguide system and send it to a system of main control to automate the mechanical system in order to optimize the time between surgeries.

The complete project consists of 4 parts distributed to 4 students. The first part focuses on the design of the mechanical structure, another on the adjustment of the stepper motors, one on the control knob and the last one on the design of the character recognition system. The focus of this graduation work is to design an Optical Character Recognition (OCR) system. An easy-to-use and user-friendly software was designed to facilitate its use, likewise for the analysis of the Brainlab Varioguide screen a web camera was used due to its price and definition of this to later do the processing of data from the images obtained, and thus find the variables of interest, which in this case are the error percentages shown on the Brainlab screen.

The Varioguide system is a system that is manually adjusted in stages, which has a ball joint for coarse adjustments and 4 joints (3 angular and 1 linear) for fine adjustments. The automation was carried out in the last 4 meetings of the system since they are the meetings that the Brainlab system shows its error percentages. When the operator is in this phase, the software automatically detects in which of the 4 engines it is at the moment to then obtain a photo, and with this image processing is carried out with matrix changes to be able to obtain the error percentage the motor.

When obtaining the value of the error percentage of a motor, a special filtering of the variables is carried out in order to verify the information. When verifying if the data is correct, the data is sent as indicated by the main control.

Los sistemas reconocimiento ópticos de caracteres (OCR por sus siglas en inglés) son una rama interdisciplinaria de la ciencia e ingeniería en sistemas que está siendo utilizada para procesos industriales, médicos, educacionales, proyectos con sistemas autónomos etc. El sistema OCR se usa con la finalidad de poder ayudar a interpretar información y usar los datos obtenidos como medio de retroalimentación de sistemas mecanizados en tiempo real. En el presente trabajo se desarrolla un software médico, en el cual se obtienen los porcentajes de error de Brainlab por medio de OCR por medio de una cámara que toma fotografías. Se procesan los datos obtenidos, y se verifica su veracidad. El enfoque de este proyecto recae en la programación y obtención de datos en tiempo real.

El Centro de Epilepsia y Neurocirugía Funcional HUMANA utiliza el sistema de navegación de Brainlab para poder ayudar con las cirugías. Este proceso es preciso por lo que requiere tiempo de preparación y estudio, por lo que HUMANA solicitó a la Universidad del Valle de Guatemala ayuda con la robotización del brazo mecánico. Este es el tercer diseño que se está realizando y tiene que contar con un manejo preciso del control del brazo asistencial. Teniendo estos datos en cuenta se propuso la idea de usar dos cámaras web para hacer reconocimiento de imágenes, pero, estas ideas fueron desechadas a lo largo de la investigación ya que no tomaba en cuenta el sistema de Brainlab. Luego se propuso la idea de diseñar e implementar un sistema OCR para la obtención de datos del sistema ya mencionado con anterioridad.

Dado a que el sistema OCR debe de ser de fácil uso se tomó la decisión de diseñar un software intuitivo amigable con el usuario, el cual controla las variables y el procesamiento de imágenes, se implementara una cámara web cerca de la posición de la pantalla de Brainlab con el propósito de tomar fotografías. Y se usara comunicación serial para enviar la información al control de los motores.

Este proceso es de gran ayuda para HUMANA ya que optimiza las cirugías, al reducir los tiempos entre operaciones. Con esto se abre un camino en el que Guatemala de un paso adelante en la innovación y desarrollo de este tipo de tecnologías ya que no se cuentan con registros nacionales sobre el desarrollo de un producto similar fabricado en el país.

Los sistemas robóticos han permitido mejorar la modificación de los procedimientos asociados a la neurocirugía mediante el uso de guías de imagen médica durante la operación. En este caso la operación se llevaría a cabo en áreas adentro del cerebro. Como el sistema ROSA de la Figura No. 1. o el sistema DaVinci los cuales son un sistema quirúrgico teleoperado de uso general. Sus características son varias como poder modificar su rango de operación a pacientes con epilepsia. pero su gran desventaja es que a gran escala el sistema solo sirve para un uso, y no puede ofrecer el acceso a todas las áreas del cerebro ya sea por huesos cercanos.[1]



Figura 1: Rosa
[1]

Neuro Arm

Este proyecto consiste en crear por primera vez una imagen guiada por MR el cual es compatible para la microcirugía. Este robot de cirugía clínica promete precisión y exactitud para que el robot pueda tomar decisiones en base a al paciente. Cuando usa la información del 3.0T (sistema de resonancia de imagen magnético operativo), el cirujano será capaz de poder observar en tiempo real, el estado de la lesión, el cerebro y la localización del aparato de cirugía. [2]

Rosa

El robot Rosa One brain es un brazo especializado para la asistencia en cirugía cerebral, este robot tiene la capacidad de incrementar la rapidez de la operación por 3 horas y 42 minutos a comparación de los modelos estándar. Además, que también cuenta con una exactitud y precisión lo hace uno de más capaces a la hora de una cirugía ya que evade las áreas más críticas del cerebro y emplea las herramientas correctas en la posición correcta para no dañar a la persona. [1]

La mecánica de la craniectomía descompresiva

Este proyecto se realiza en el centro de investigación de Stanford University. El trabajo presenta una craniectomía de modelo computacional que ayuda para cuantificar el cambio del cerebro, este estudio ayuda a poder identificar los grandes riesgos de regiones que son propensas a daños cerebrales durante una cirugía. [3]

Un método combinado de registro y análisis de elementos finitos para una estimación rápida del desplazamiento cerebral intraoperatorio

Este proyecto se realiza en el centro de investigación universidad de Teherán. El estudio propone una combinación de registro de información preoperatorio e intraoperatorio. Utiliza algoritmos optimizados de un software de análisis de elementos finitos, lo que permite acortar los procedimientos de cálculo de la deformación volumétrica. Los autores del estudio esperan que el método propuesto acelere el procedimiento general de estimación del desplazamiento cerebral. [4]

Evaluación clínica de un modelo de imagen actualizada enfoque de orientación para compensación de cambio de cerebro

El estudio se realizó en el centro de investigación Vanderbilt. En este estudio, los autores evalúan la robustez y precisión de un modelo biomecánico para la corrección del desplazamiento cerebral que se desarrolló para la cirugía de resección de tumores. [5]

La epilepsia es la responsable de una proporción significativa de la carga mundial de morbilidad, ya que está afecta a más de 50 millones de personas. la proporción estimada de la población general con epilepsia activa es 4 y 10 de cada 1000 habitantes. En Guatemala se considera un problema social ya que esta afecta alrededor de 325,000 personas, lo que conlleva a tener un alto impacto emocional, personal, familiar, social, y en muchas ocasiones el problema no son las convulsiones, si no el lugar en la que se dan, ya que puede poner en riesgo su vida y sus alrededores. Aproximadamente de estas 325,000 personas con epilepsia el 0.3 % de 1 % de los pacientes con epilepsia es de difícil control.[6]

Los datos en Guatemala muestran que 7 de cada 10 personas con epilepsia que se presentan para tratamiento con medicamentos son capaces de llevar una vida normal libre de epilepsia y de más tratamientos, pero 3 de cada 10 personas tienen que recurrir a cirugías, para la cual en muchas ocasiones la epilepsia es curable.

El brazo biomédico de electroshocks como en la Figura No.2. es el avance de los procedimientos para un tratamiento más efectivo ante la epilepsia. Sería un avance para el país usarlo y tratar este tipo de enfermedades, al poder aumentar la precisión y exactitud de la operación se podrá mejorar los tratamientos y procedimientos antes estas operaciones y emergencias. Y no solo esto, el brazo biomédico aumentará la velocidad de los procedimientos lo que apoyará con el tiempo a los centros médicos en toda Guatemala. Con esto se abre las puertas de oportunidades para avances más tecnológicos, y ayudará significativamente a 350,000 guatemaltecos, lo que representa un gran reto para nuestro equipo y la Universidad del Valle de Guatemala.



Figura 2: Operación cerebral
[1]

4.1. Objetivo general

Diseñar un sistema de procesamiento de imágenes con una cámara web que lea las variables de calibración de la pantalla del sistema de Brainlab del Centro de Epilepsia Y Neurocirugía Funcional HUMANA.

4.2. Objetivos específicos

- Diseñar y fabricar un dispositivo de acople para una cámara web a la pantalla de Brainlab del centro de epilepsia y neurocirugía funcional HUMANA.
- Implementar un sistema de procesamiento de imágenes y caracteres capaz de identificar la junta que esta siendo calibrada y el error desplegado.
- Implementar un protocolo de comunicación de envío de datos al sistema de control principal.

La epilepsia es una enfermedad cerebral no transmitible crónica que afecta a personas de todas las edades, y afecta a 50 millones de personas alrededor del mundo. Con un tratamiento anticonvulsivante para más de la mitad de la población mundial y la cirugía puede ser conveniente en casos en los que los pacientes no responden a los tratamientos farmacológicos.

La OMS y sus asociados reconocen que la epilepsia es un importante problema de salud pública. Los esfuerzos médicos han contribuido a darle una alta prioridad a la enfermedad. Y en muchos países se están llevando proyectos para reducir la falta de tratamiento y la morbilidad de las personas con epilepsia, capacitar y educar a los profesionales sanitarios, e identificar posibles estrategias. Como se lleva a cabo en HUMANA la cual es el Centro De Epilepsia Y Neurocirugía Funcional en Guatemala. El centro usa tecnología de vanguardia guiada por su programa llamado Varioguide el cual ayuda a los médicos a poder aumentar el rendimiento de la operación y acortar la duración de este. Gracias a los esfuerzos del centro se han tratado cientos de personas en el área metropolitana y rural del país. Con el proyecto de identificación de variables se agilizará el proceso de operación, así como el rendimiento de este, aumentando el alcance del centro y poder expandirse a más localidades del país y así poder ayudar a más personas que requieran ayuda en Guatemala.

El proyecto se enfoca en la implementación de un sistema de Reconocimiento ópticos caracteres (OCR por sus siglas en inglés) para un brazo robótico que se empleará en conjunto con Brainlab. Este brazo asistencial se hizo en colaboración con Héctor Cano, Peter Yau, y Rodrigo Corona. El brazo debe ser robusto, estable, permitir su uso con Varioguide, y que cuente con la retroalimentación del sistema OCR. Este brazo asistencial cuenta con 8 grados de libertad, pero solo los últimos cuatro podrán tener la capacidad de moverse automáticamente.

El sistema que se desarrolló contará con el software de reconocimiento óptico de caracteres, procesará las imágenes obtenidas por la cámara web, para buscar los datos importantes haciendo cambios matriciales a los píxeles, y buscará áreas de interés. Ya obteniendo los datos se hace un filtrado de datos para asegurar que lo obtenido concuerde con lo que se tiene que enviar eliminando la información no relevante. Con el dato procesado se prosigue

enviándolo a un sistema de control principal de motores.

En el sistema OCR hace falta la implementación de la cámara web, con su base de ajuste manual, que tenga la capacidad de ser firme y que pueda soportar cargas puntuales. Esta base tiene que estar siempre en la misma posición, de lo contrario las áreas obtenidas por el sistema OCR ya no serían las mismas y los datos obtenidos cambiarían si se llega a afectar la imagen de la cámara. El prototipo debe contar con estos ajustes para que el sistema esté finalizado.

6.1. La epilepsia

La epilepsia es un trastorno cerebral el cual provoca convulsiones repetidas durante un tiempo. Las convulsiones son episodios de actividad descontrolada y anormal de las neuronas que puede causar cambios de atención o el comportamiento.

Las causas de la epilepsia son variadas y mayormente pueden ser producidas por cambios en el tejido cerebral que hacen que el cerebro este excitable o irritable. A causa de esto el cerebro envía señales anormales, lo que ocasiona convulsiones las cuales en el mayor de los casos son impredecibles.

Esta puede deberse a causa a una afección de salud o una lesión en el cerebro, y en varios casos son desconocidos. [6]

Las causas comunes de la epilepsia pueden ser :

- Accidente cerebrovascular.
- Demencia.
- Infecciones como meningitis , encefalitis, VIH, etc.
- Problemas cerebrales al nacer.
- Lesión cerebral durante el nacimiento o después del mismo.
- Trastornos metabólicos presentes al nacer.
- Tumor cerebral.
- Vasos sanguíneos anormales.

- Trastornos epilépticos.

Estas crisis anormales empiezan después de los 5 a 20 años, también hay una probabilidad en adultos mayores de 60 años. [6]

6.2. Visión por computadora

La visión artificial o mejor conocida como visión de computadora o visión técnica, es un estudio científico que incluye métodos para adquirir, procesar, analizar, y comprender las imágenes del mundo real con el fin de producir información numérica y simbólica para que puedan ser tratados por un ordenador, así como los humanos usamos nuestros ojos para ver, este sistema tendrá sensores y su propia visión artificial para poder interpretar la información.

La visión artificial trata de producir el mismo efecto para que los ordenadores puedan comprender una imagen o secuencia de imágenes y actuar a conveniencia en una determinada situación. [7]

6.3. Python

Python es un lenguaje de programación interpretado cuyo propósito se enfoca en su código propio y único para programar o codificar acciones. Se trata de un lenguaje multiparadigma, ya que soporta parcialmente la orientación de objetos, programación imperativa, y programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

Python 1.0 es una versión creada en 1994. Una característica de este lanzamiento fueron las herramientas de la programación funcional, las cuales tenían la función de lambda, reduce, filter y map. Python empezó con funciones simples y fue creciendo hasta el Python que se conoce hoy en día con varias funciones librerías y aplicaciones para utilizar Python. [8]

6.4. Reconocimiento óptico de caracteres (OCR por sus siglas en inglés)

El Reconocimiento óptico de caracteres (OCR por sus siglas en inglés) es un tipo de tecnología que trata de emular la capacidad del ojo humano para poder reconocer objetos. Concretamente este software tiene la capacidad de reconocer por medio óptico los caracteres de una imagen, de forma que estos se vuelven comprensibles o reconocibles para cualquier tipo de ordenador, y tiene como finalidad obtener un archivo de texto editable. [9]

6.5. Tesseract

El Tesseract es el motor de OCR el cual es público que desarrollo la empresa de HP entre 1984 y 1994. Con junto la biblioteca de procesamiento de imágenes de Leptonic este puede tener la capacidad de leer una gran variedad de formatos de imágenes y convertirlos en texto.

Para poder descifrar una imagen adecuada con OCR, y que envíe un texto correcto, se necesitan las siguientes características. [10]

- Una imagen binaria (Imagen blanco y negro).
- Del tamaño de la imagen no debe ser grande.
- Se debe delimitar el área de interés.
- Se tiene que resaltar el texto en el fondo.
- Se debe de evitar ruido, y eliminar pixeles que no forman parte del texto.
- La tipografía debe ser común, como Arial, Roman, o Tahoma.

6.6. Matlab

Matlab es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio. El cual está disponible para las plataformas de Unix, Windows, Macos y GNU/Linux.

El cual en sus funciones básicas tiene la capacidad de utilizar matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario y la comunicación con programas de otros lenguajes y con otros dispositivos hardware. Este programa dispone de varios paquetes, los cuales contienen herramientas adicionales que expanden las prestaciones, como Simulink y GUIDE. Y no solo esto el programa tiene la capacidad de ampliar sus capacidades usando estos paquetes extras.[11]

6.7. BrainLab

Brainlab es una entidad privada alemana de tecnología médica el cual tiene sede en Munich ,Baviera. Brainlab se enfoca principalmete en desarrollar software y hardware para radioterapias, y radiocirugías , y los campos de neurocirugía, otorrinolaringología y craneo-maxiolfacial, cirugía de columna e intervenciones traumáticas. Tienen productos que varían desde intefración a quirófanos digitales e intercambio de datos en la nube. [12]

6.8. Pixel

Los pixeles son los puntos de color de una escala de grises a una escala monocromática, estas forman las imágenes por medio de una sucesión de las mismas, esta sucesión marca la coherencia de la información presentada, siendo su conjunto una matriz coherente para el uso digital. El área donde estas se proyectan es normalmente rectangular. La representación del pixel en pantalla, al punto de ser accesible por unidad, forma un área homogénea en cuanto a la variación del color y densidad de pulgada, siendo esta variación nula, y definiendo cada punto sobre la densidad, en lo referente al área.

En las imágenes de mapa de bits. Cada pixel se puede codificar con un conjunto de bits de longitud determinada , puede codificarse con un byte , de manera que cada pixel admite 256 variaciones de color, de 0 a 255.[13]

6.9. Cámaras digitales

Las cámaras digitales usan componentes de electrónica que son fotosensibles, como los CCD o sensores CMOS, los cuales graban a niveles de brillo en una base por pixel, en la mayoría de las cámaras digitales el CCD este cubierto por un mosaico de filtros de color, teniendo regiones de color rojo, verde y azul (RGB por sus siglas en inglés), organizados normalmente en filtro de Bayer, por lo que el sensor solo puede grabar el brillo de un solo color primario. La cámara se encarga de interpolar la información de color de los pixeles vecinos, mediante un proceso llamado de interpolación cromática para crear la imagen final.[14]

6.10. Dispositivo de carga acoplada (CCD por sus siglas en inglés)

Un dispositivo de carga acoplada es in circuito integrado que contiene un número determinado de condensadores enlazados o acoplados, los CCD se encuentran normalmente entre las cámaras digitales profesionales. Estos dispositivos fueron inventados por Willard Boyle y George E. Smith el 17 de octubre de 1969 en los laboratorios de Bell. La capacidad de resolución de la imagen depende directamente del número de células fotoeléctricas del CCD. Este numero se expresa en pixeles. A mayor número de pixeles, mayor nitidez en relación con el tamaño.

Los pixeles del CCD registran graduaciones de los tres colores primarios, rojo verde y azul, por lo cual tres pixeles, uno para cada uno forma un conjunto de células fotoeléctricas capaces de captar cualquier color de la imagen.

Para poder conseguir la separación entre cada color se utiliza una máscara de Bayer la cual proporciona una trama para cada conjunto de cuatro pixeles de forma que un píxel registra luz roja, otra luz azul y dos pixeles de reservan para el verde.[15]

6.11. Mosaico de Bayer

Una cámara digital tiene la capacidad de poder usar millones de pequeñas cavidades de luz llamadas fotositos para poder capturar la imagen.

Cuando se presiona el obturador la exposición comienza, cada uno de estos este descubierto para poder recolectar fotones y guardarlos como señales eléctricas, una vez que la cámara termina la exposición, cierra cada fotosito, para luego poder identificar cuantos fotones están en cada cavidad para medir la intensidad de la señal eléctrica.

La señal es cuantificada en señales digitales, su precisión depende por la profundidad del bit, lo que resulta en una precisión reducida en formato guardado de 0-255 por 8-bit en formato JPEG. Se puede observar el comportamiento en la Figura No. 3.

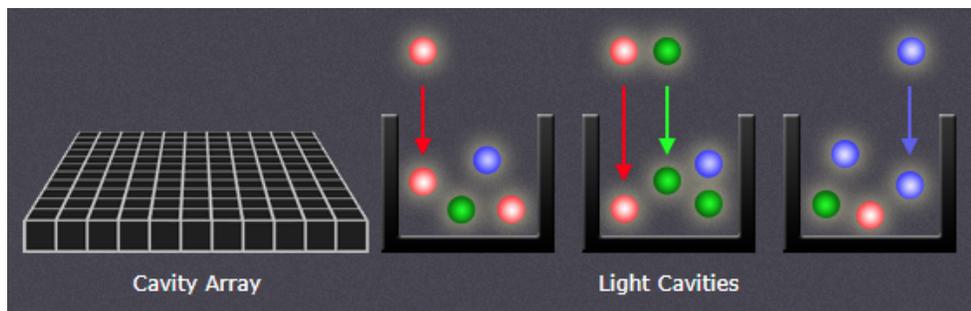


Figura 3: Cavidades de luz

Sin embargo, la ilustración solo crea imágenes en escala de grises a razón que las cavidades no distinguen cuánto hay de cada color. Para poder capturar imágenes a color se tiene que colocar un filtro en cada cavidad que deje pasar fotones de ciertos colores.

Virtualmente se pueden capturar todos los colores obteniendo los tres colores primarios en cada cavidad, y en cada cavidad solo un color primario puede pasar, por lo que $2/3$ de la luz proveniente se descarta. Como resultado la cámara tiene que aproximar el resto de los dos colores para poder recrear el píxel.

Un filtro de Bayer consiste en arrays de rondas de rojos-verde y verde-azul. Cada color primario no recibe una fracción igual del área total ya que el ojo humano es más sensible a la luz verde a comparación de la roja o azul. Con esto se reduce el ruido con el color verde ya que se pueden dejar más detalles solo con este color. Se puede observar el comportamiento en la Figura No. 4. [16]

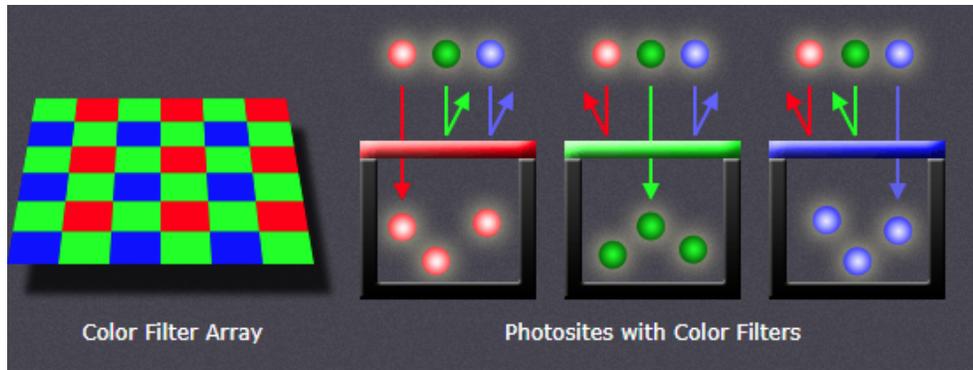


Figura 4: Filtro Bayer

6.12. Rojo Verde Azul (RGB por sus siglas en inglés)

RGB significa Rojo, verde, y Azul, se refiere a tres tonos de luz que se pueden mezclar para crear diferentes colores. La combinación de luz roja, verde y azul es el método estándar para poder reproducir imágenes en color en las pantallas, como en televisores, computadoras, etc. El número de colores admitidos por RGB depende de cuantos valores posibles se puedan usar para rojo, verde y azul. Esto se conoce como “profundidad de color” y se mide en los bits. La profundidad de color más común es el color 24 bit, el cual admite un total de 24 bits, lo que proporciona 256 posibles valores para rojo, verde y azul o 16,777,216 colores posibles en la paleta de color con la combinación de los 3 colores. [17]

6.13. Zoom digital

Cuando se trabaja con un zoom únicamente digital nos podemos acercar a una escena sin necesidad que la lente realice ningún trabajo mecánico, ni haya vidrios que se desplacen, esto es porque lo que hace es ampliar la imagen y realizar un recorte de un área para que parezca que está más cerca.

Cuando se toma una imagen con zoom digital lo que se hace es reencuadrar la foto en un programa de edición, pero con la diferencia que la parte restante se rellenara gracias a una extrapolación de pixeles nuevos creados a partir de otros colindantes que se tomarán automáticamente como referencia.

Por lo tanto, si se obtiene un sensor de 8x8 pixeles y se realiza un zoom digital se usará una porción del sensor de 4x4 pixeles y se colocaran pixeles intermedios falsos creados por algoritmos complejos que, según lo avanzados que sean determinaran la calidad del resultado. Se puede ver el zoom digital en la Figura No. 5. [18]

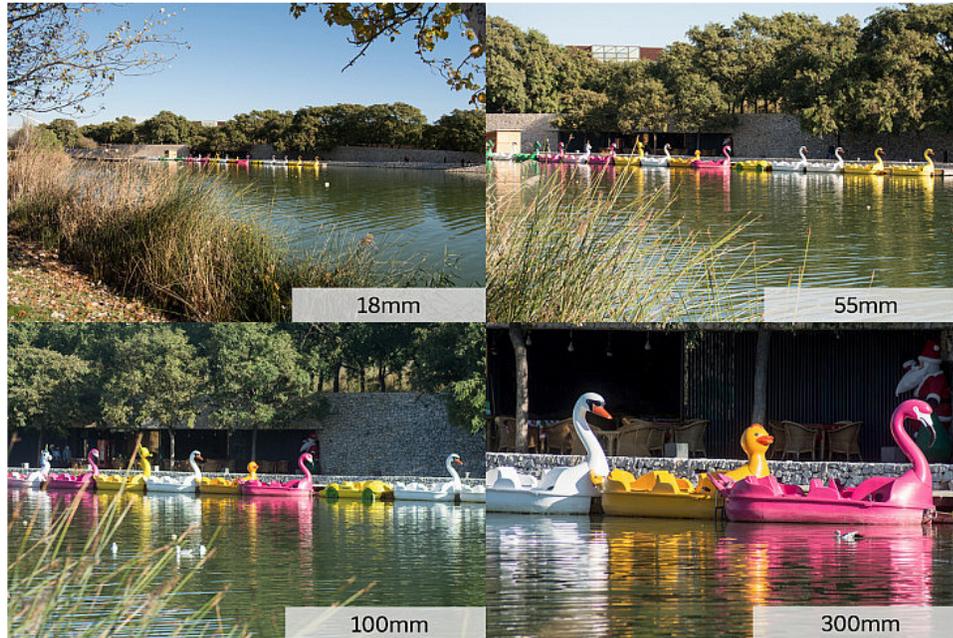


Figura 5: Zoom digital

6.14. Zoom óptico

Un zoom óptico tiene la misma funcionalidad que un zoom tradicional del zoom óptico de un objetivo, el vidrio cambia de posición dentro de la lente manipulando los rayos de luz que penetran hasta llegar al sensor para acercar o alejar las escenas que se observan.

Este zoom afecta a la función de la distancia que haya entre el lente y el sensor, la imagen resultante será mayor o menor. Esta distancia entre la lente y el sensor es lo que se denomina distancia focal.

En resumen, a cuanto mayor sea la distancia focal y más lejos este la lente del sensor más aumentado veremos la imagen, teniendo un ángulo de visión de las escenas más cerrado. Como se puede observar en la siguiente imagen con un zoom de 200 mm nos acercaremos más y con 18 mm veremos la escena más alejada. En la Figura No. 6 se puede ver el zoom óptico.[19]

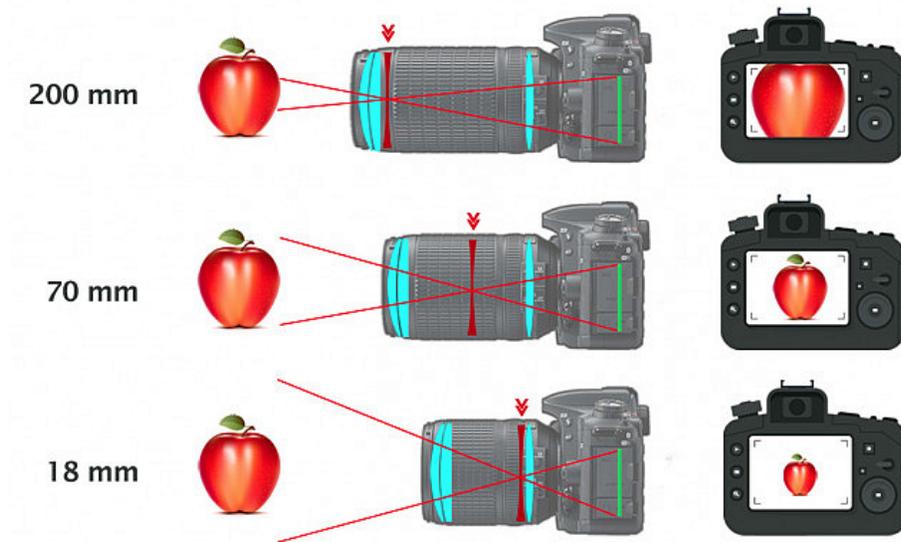


Figura 6: Zoom óptico

6.15. Lentes lineales

La mayoría de aplicaciones para lentes lineales se pueden presentar en lentes, espejos, placas de onda, etc. Se dice que es un lente lineal si cumple con las siguientes descripciones:

- Si la luz monocromática entra sin cambiar el sistema óptico lineal, y la salida tiene la misma frecuencia que entrada.
- El principio de superposición se debe de cumplir y validar en un sistema óptico lineal, esto quiere decir que, si el espejo transforma la entrada A a la salida B, y la entrada C a la salida D, la entrada A y C tienen que dar simultáneamente la salida B y D.
- Si la entrada de la luz es más intensa, la salida tiene que ser más intensa, pero sin cambiar la imagen. [20]

6.16. Sensibilidad ISO

Una exposición fotográfica depende de tres variables, apertura, velocidad de obturación y sensibilidad ISO. Para los fotógrafos que usan películas, la apertura y la velocidad de obturación pueden cambiar de un cuadro para otro, pero la sensibilidad ISO es siempre constante, por lo menos hasta que se termine el rollo de la película.

ISO es la sensibilidad a la luz de una película o de un sensor de Imagen. Una ISO baja necesita más la luz para lograr una buena exposición, mientras que una ISO alta es más sensible a la luz y necesita menos luz para una buena exposición. Al disparar a plena luz

del día lo más recomendable es usar una sensibilidad ISO baja. Al disparar con poca luz, un incremento en la sensibilidad ISO hace que el sensor de la cámara sea mas sensible a la falta de luz en escena , pero al aumentar el ISO algunas veces se incrementa el ruido o la granulación de la imagen. El comportamiento se puede verificar en la Figura No. 7. [21]



Figura 7: Ajuste de balance blanco.

6.17. Temperatura de color

La temperatura de color describe el espectro de luz que irradia de un cuerpo negro con superficie de temperatura. Un cuerpo negro es un objeto que absorbe la luz. Este fenómeno se puede ver en materiales como metal, cuando es calentado se dice que se está poniendo rojo, y un rojo blanco cuando alcanza altas temperaturas, un cuerpo negro a diferentes temperaturas puede variar de color y puede tener diferentes distribuciones de color. Como se puede observar en la Figura No. 8. [22]

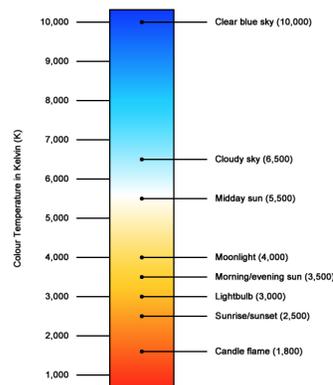


Figura 8: Temperatura del color

6.18. Sharpness

El sharpness o nitidez describe la claridad del detalle de las fotos o imagen. Con el uso de técnicas de procesamiento de imágenes se puede llegar a improvisar la nitidez de la imagen. Pero esta normalmente esta limitada por la cámara o uso de algún dispositivo como televisor o computadora, la imagen incrementada o la distancia focal. Los principales factores que contribuyen a la nitidez de la imagen son:

- Acutancia
- Resolución

6.19. Acutancia

La Acutancia describe que tan rápido se puede trasladar la información en las esquinas de la imagen, normalmente las imágenes que son claras y bien definidas tienen los bordes claros. Se observa el comportamiento en la Figura No. 9.



Figura 9: Acutancia

6.20. Resolución

Describe la habilidad de la cámara para distinguir entre el espacio de los detalles en los elementos. La resolución se puede observar en la Figura No. 10.



Figura 10: Resolución

Sistema Varioguide de Brainlab

El sistema Varioguide de Brainlab es una sinergia entre un sistema de control por imágenes y un sistema mecánico de operación manual. El operario del sistema debe ajustar diferentes juntas mecánicas de forma manual de acuerdo con lo que el sistema de Brainlab le indique en una pantalla como en la Figura No. 11. Para simplificar la tarea del operario se implementó un sistema de visión por computadora que lee las variables directamente de la pantalla y luego las transfiere a un sistema de control automático. El sistema de control se encarga de mover las juntas físicas por medio de motores.

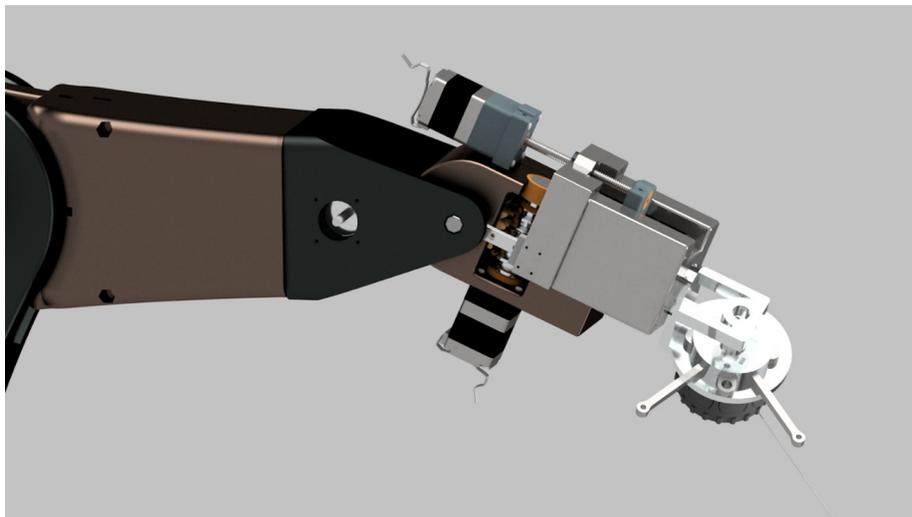


Figura 11: Diseño de brazo mecánico

El brazo mecánico propuesto utiliza 8 motores para el control del efector final. Los primeros cuatro motores son controlados manualmente y los últimos cuatro motores son

controlados de forma manual o de forma automática. El sistema de visión por computadora implementado en este trabajo de graduación permite leer la pantalla de Brainlab, las juntas o el motor que requiera ajustes como se puede ver en la Figura No. 12, así como el error que existe. Una vez que se hayan identificado el motor y error indicados en la pantalla, el software se encarga de enviar estos datos al sistema de control que moverá los motores.

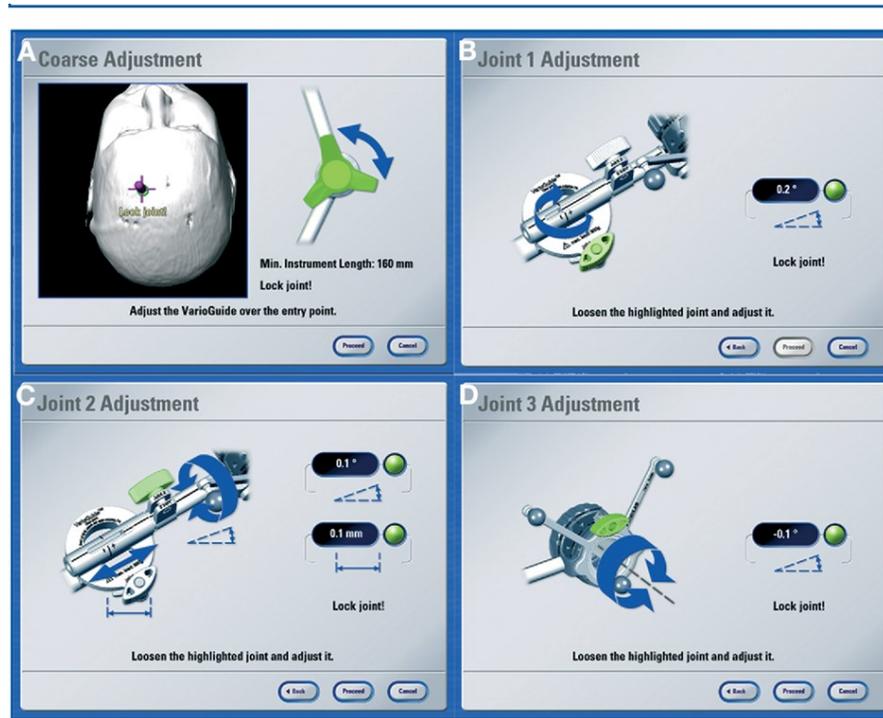


Figura 12: Pantallas de ajustes en Brainlab

7.1. Nuevo prototipo

En el diagrama presentado en la Figura No. 13 se puede observar la interacción entre la interfaz y el usuario. El nodo principal el cual es el sistema que controla a los 2 nodos de interfaz manual e interfaz automática, el usuario podrá seleccionar una de las 2, para este estudio solo se utilizará el modo de interfaz automática el cual es el que nos interesará. En este modo el usuario tendrá 4 opciones para poder verificar los motores 5,6 ,7 y 8 , si el sistema encuentra cualquiera de los 4 podrá enviar los datos del software por conexión UART al sistema principal, estos datos provienen del porcentaje de error de los mismos motores y describen el ángulo para poder ubicar el brazo asistencial.

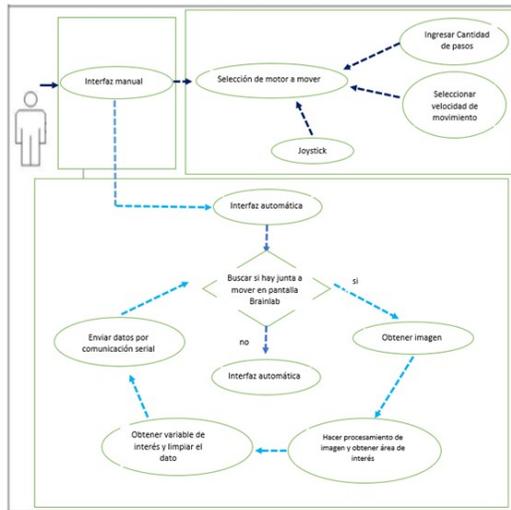


Figura 13: Diagrama de flujo del programa capaz de obtener datos de las pantallas de Brainlab

7.1.1. Interfaz principal

Como el proyecto solo se enfocó en el modo interfaz automática y no se tomó en cuenta el sistema del movimiento de los motores, como podrán observar en la Figura No. 14, solo se tomó en cuenta las imágenes de los motores, la información del motor y su error, su verificación, procesamiento de imágenes, la búsqueda del área del cambio de error así como la variable de interés y el envío del dato obtenido de esta misma a un sistema de control de motores.

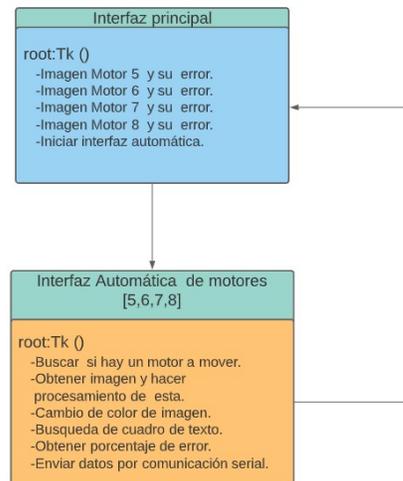


Figura 14: Interfaz automática

7.1.2. Clases del sistema

En la Figura No. 15 se puede observar las clases del sistema y sus dependencias que existen entre cada una de ellas. El sistema principal llamado interfaz principal tiene los métodos e instrucciones para mostrar la interfaz principal y permite llamar otra clase de acuerdo a la sección del usuario que visualiza el sistema Automático. Para poder manejar los ángulos y el desplazamiento de los motores el usuario puede observar los datos mandados, y todo por medio de interfaces gráficas. Se implemento el sistema en Tkinter de Python para poder hacer las interfaces gráficas.

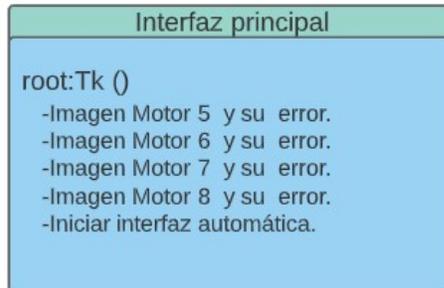


Figura 15: Interfaz principal

Para la aplicación principal se tomó en cuenta que aparecieran los 4 de los 8 motores para el sistema automático como se puede identificar en la Figura No. 16, en el cual al momento de identificar el error este aparezca en la pantalla de la aplicación y se pueda observar en tiempo real lo que el OCR identifica.

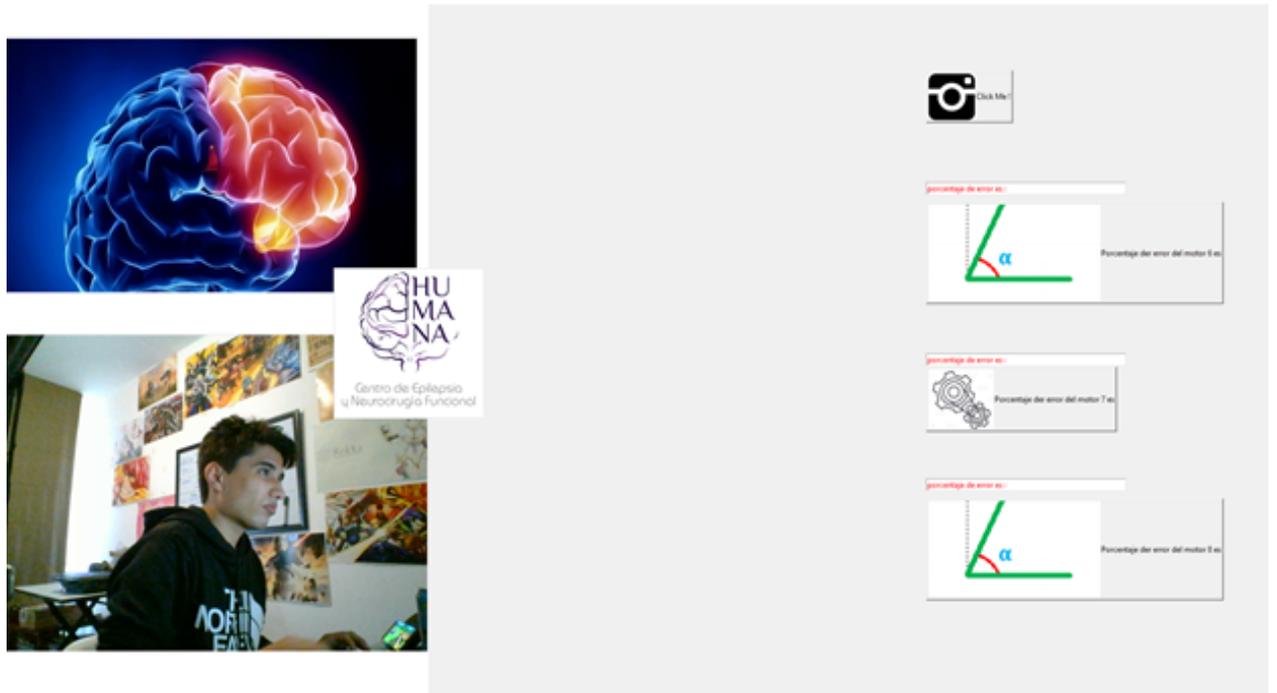


Figura 16: Aplicación del sistema principal

7.1.3. Interfaz automática

La interfaz automática como se puede verificar en la Figura No. 17 está conformada por 7 métodos, en esta interfaz se puede obtener la imagen de la cámara web y las constantes de los 4 motores. El usuario puede observar que datos y a que motor pertenece el valor del porcentaje de error. Además, en esta interfaz se encuentra el sistema de reconocimiento de OCR el cual verifica la pantalla y si encuentra una pantalla de interés de uno de los 4 motores hace un preprocesamiento de imagen obteniendo el porcentaje de error sin importar la posición de la ventana emergente.

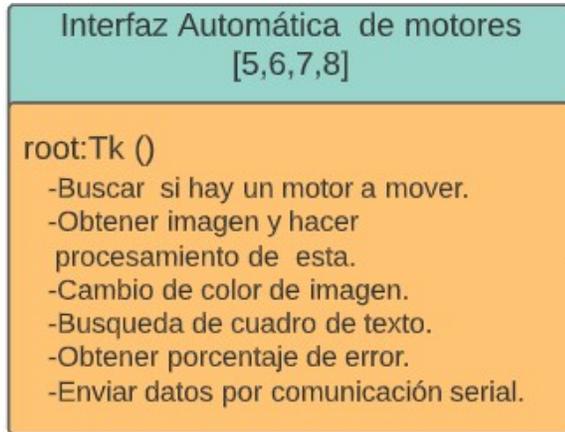


Figura 17: Métodos de interfaz automática

Se puede observar en la Figura No. 18 la búsqueda del área del porcentaje de error una vez que la interfaz principal haya verificado si hay imagen de Brainlab en la cámara web.

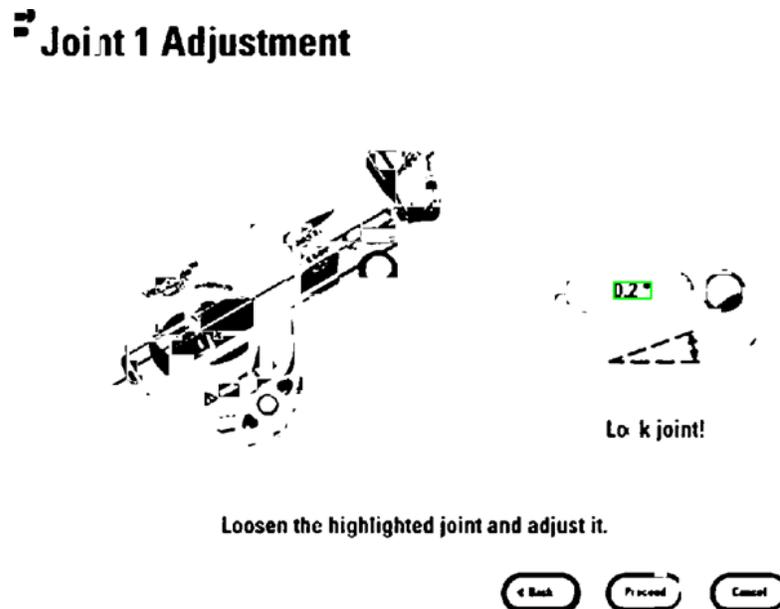


Figura 18: Búsqueda de área de interés obtenida mediante el sistema principal

Sistema de visión por computadora

El sistema de visión de computadora cuenta con siete fases como se puede observar en la Figura No. 19, estas fases dependen principalmente de la calidad de la imagen y el procesamiento adecuado para obtener datos de interés.

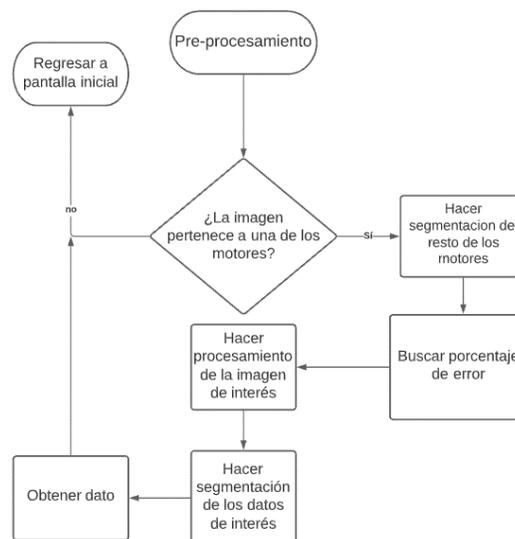


Figura 19: Sistema de reconocimiento óptico de imágenes

La etapa más importante del sistema es el reconocimiento óptico de imágenes (OCR por sus siglas en inglés) y su función es poder identificar el porcentaje de error de los motores con imágenes de las webcams como se puede observar en la Figura No. 20, este segmentó es

la clave del trabajo ya que el sistema depende del dato que se encuentre.

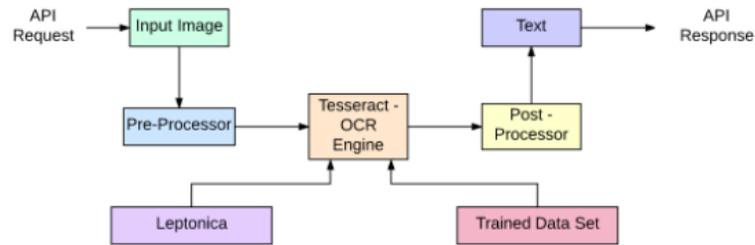


Figura 20: Reconocimiento óptico de imágenes

Esta es una de las etapas más importantes, ya que se puede hacer una segmentación si se logra identificar a que pantalla pertenece la imagen. Luego de esta segmentación se hace otra para poder separar los datos relevantes, esta función es necesaria para poder obtener los porcentajes de error de un solo motor.

La imagen puede ser formada de distintos orígenes, una de ellas es a través de la fotografía como se puede observar en la Figura No. 21. El procesamiento de imágenes se puede generar por medio de métodos ópticos y digitales.

Teniendo en cuenta a Fourier afirma que la función, se puede representar con una alta precisión dentro de un intervalo dado, mediante la suma de una familia de funciones sinusoidales, pero con diferentes frecuencias. Así es como por ejemplo la variación de la irradiación o brillo de una imagen, medida a lo largo de una dirección cualquiera es entonces una función que se representa con suma de distribuciones sinusoidales de varias frecuencias.

La calidad de la imagen puede ser mejorada dependiendo de los componentes sinusoidales de la imagen a tratar.

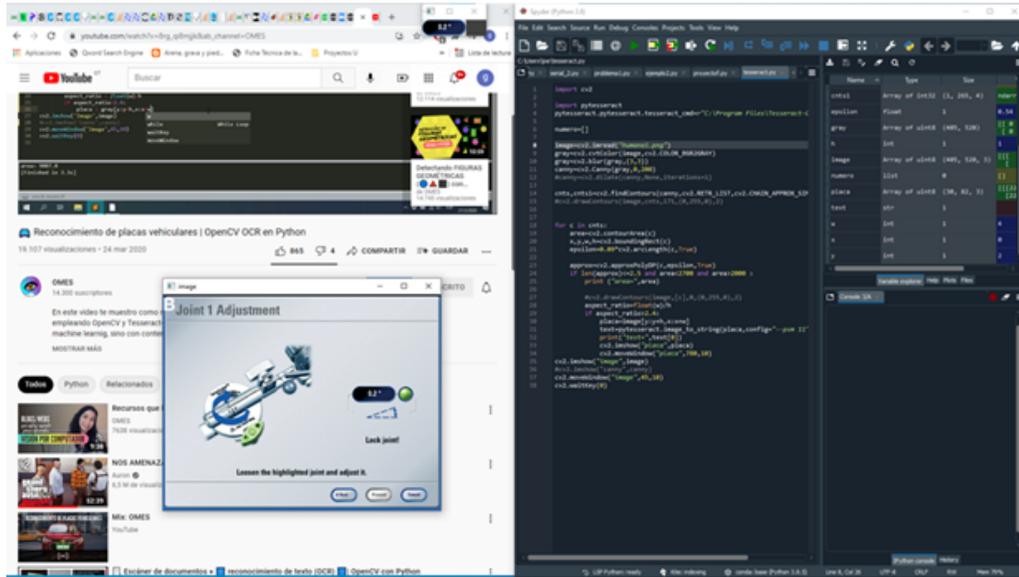


Figura 21: Obtención de captura de imagen

Para comenzar la imagen que se estaría configurando sería como la de la Figura No. 22. Se puede observar que el porcentaje de error es un ovalo. Este ovalo tiene fondo morado y letras en blanco. Para poder obtenerlo se hizo un procesamiento de imágenes pasando la imagen por 4 filtros.

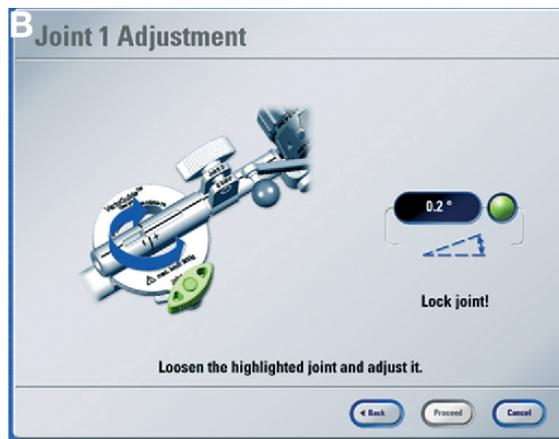


Figura 22: Imagen obtenida de Brainlab

8.1. Determinación de junta a manipular

Empieza el proceso del sistema obteniendo la imagen de la pantalla de Brainlab con la webcam. El sistema estará capturando video en todo momento, se guardan las imágenes de este proceso como se puede observar en la Figura No. 23, las imágenes de 24 bits a color pasan a ser de ocho bits en escala de grises como se puede observar en la Figura No. 24.

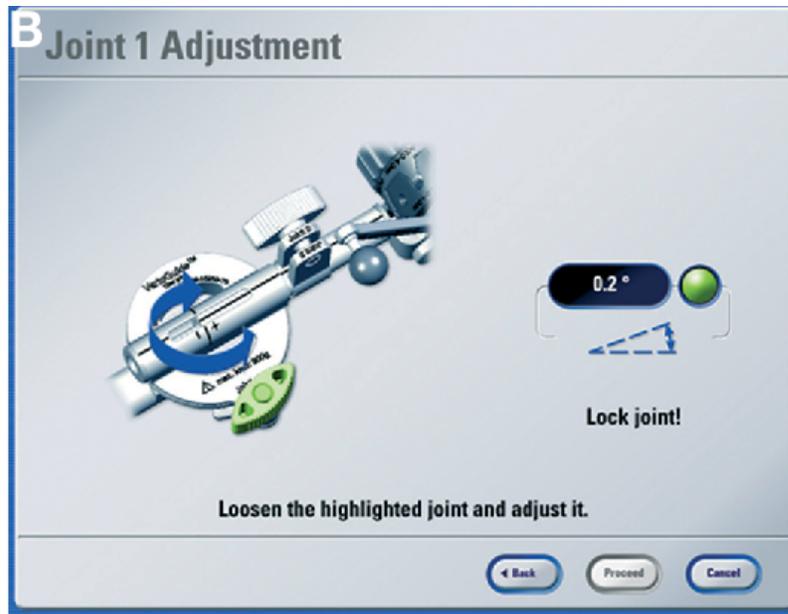


Figura 23: Imagen escala a color de 24 bits

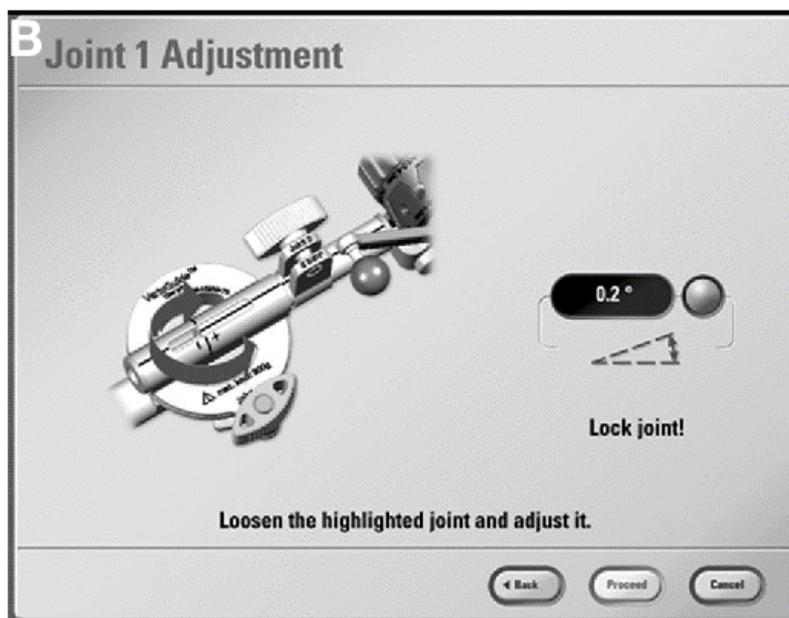


Figura 24: Imagen escala a gris de 8 bits

Con la imagen en gris se usa la función de pytesseract para usar la librería OCR en configuración 11 para obtener la mayor cantidad de caracteres que se puedan encontrar, como se puede observar en la Figura No. 25, y guarda todos los strings en una variable de texto.

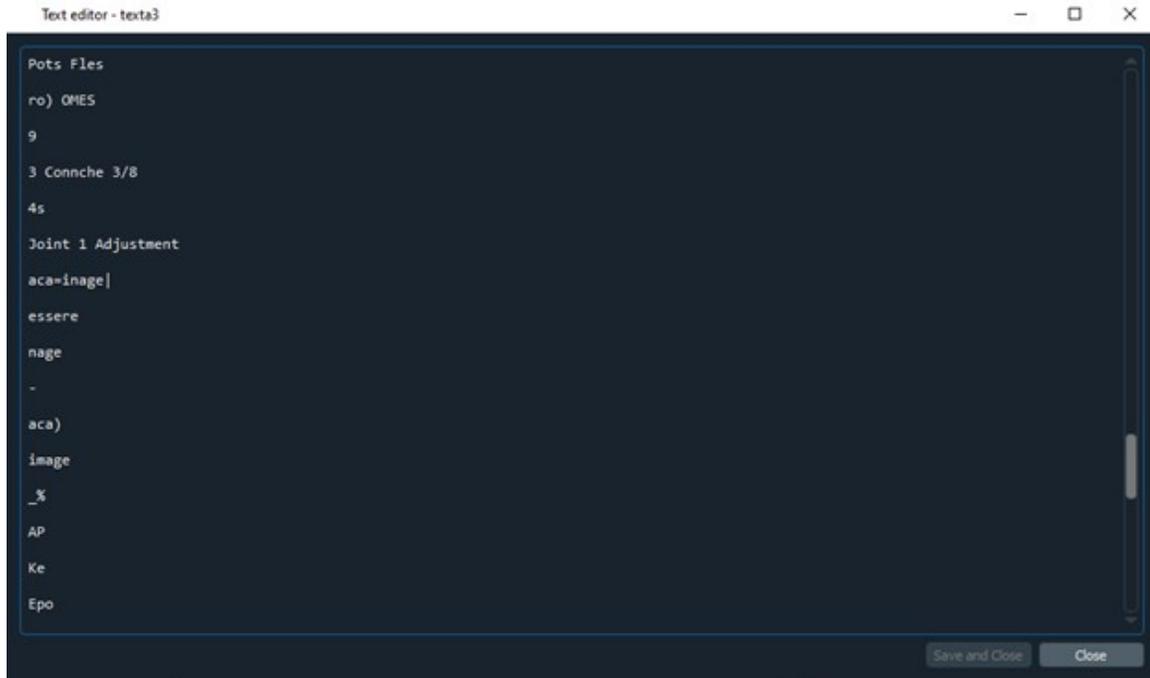


Figura 25: Buscar variables con OCR de Tesseract

Luego se verifica en todo momento si el sistema encontró los strings de joint 1, joint 2, joint 3 usando la función `.find` la cual busca cualquier string sobre algún texto. Al momento de encontrar cualquiera de las tres juntas se procede a buscar el área de interés de esta, como se puede observar en la Figura No. 26.

```
In [1]: runfile('C:/Users/joe/tesseract.py', wdir='C:/Users/joe')
632
```

Figura 26: Encontró el string correcto en la posición 632

8.2. Filtrado de color

En esta etapa se toma la imagen de la Figura No. 24 en escala de grises que se obtuvo anteriormente y se filtra por un `threshold`, este filtro necesita una imagen con escala de grises para poder funcionar, este filtro se usa para poder clasificar los píxeles en la imagen, para poder maximizar el valor de los píxeles, hay varios filtros en CV2, en mi caso se utilizó `cv.THRESHBINARY` con `cv.THRESHOTSU`, se usa esta variación para poder maximizar el valor de los píxeles que están en blanco y negro, además para que obtenga el valor del `threshold` automáticamente, como pueden observar en la Figura No. 27 y 28, la imagen obtenida se pueden identificar bien los blancos y los negros, ya que el valor de estos se amplifican. Se hace este procedimiento ya que el reconocimiento óptico de caracteres (OCR por sus siglas en inglés) solo puede identificar este tipo de escala.

$$dst(x, y) = \begin{cases} \text{maxValue} & \text{if } src(x, y) > T(x, y) \\ 0 & \text{otherwise} \end{cases}$$

Figura 27: Uso de filtro threshold

La imagen calcula el valor máximo de los pixeles y los basa a valores binarios con una imagen de blanco y negro de 8 bits.

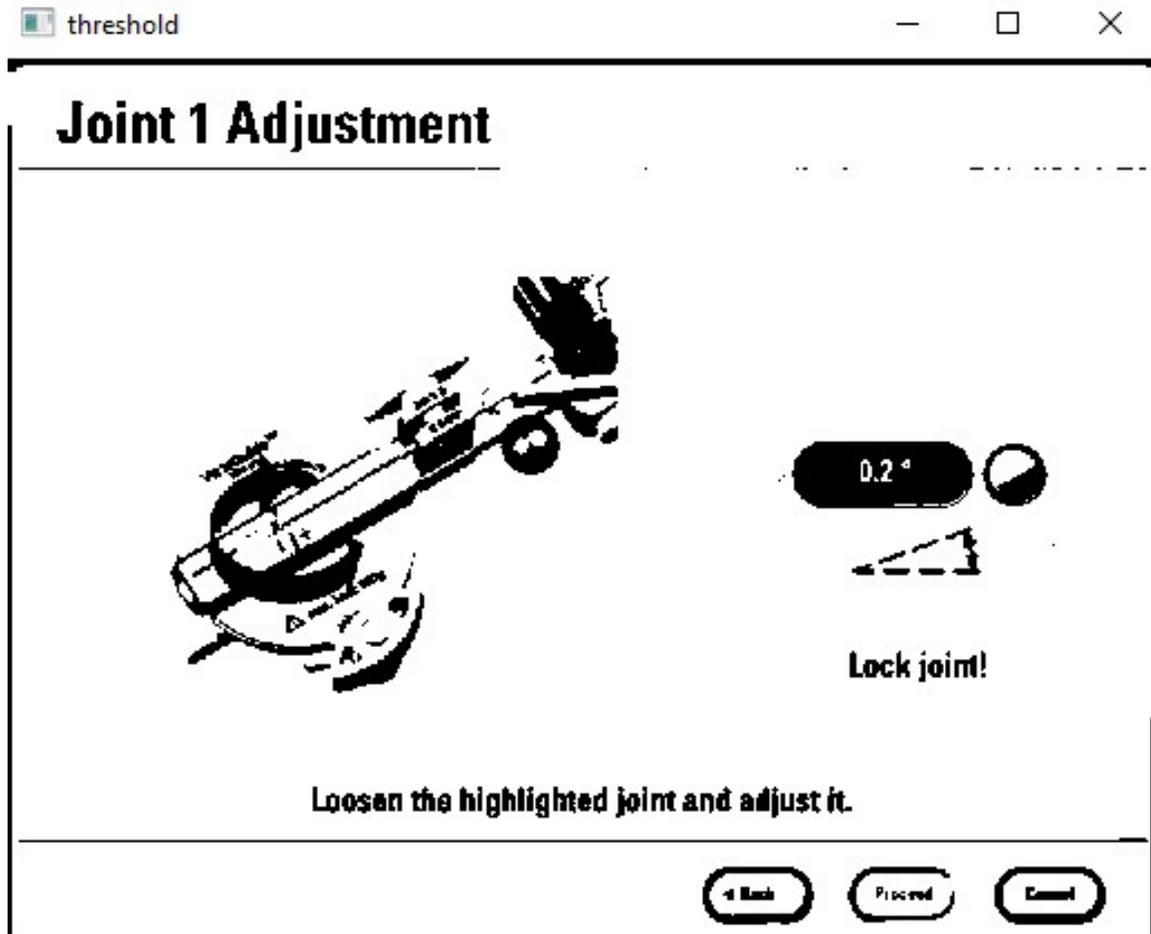


Figura 28: Imagen con filtro threshold

8.3. Encontrando el área de interés

Para encontrar el área de interés donde se encuentra el porcentaje de error de la junta se necesita usar el filtro canny. Este filtro es basado en la primera derivada de una gaussiana. Ya que las imágenes procesadas son susceptible al ruido presente en datos sin procesar, se transforma la imagen con un filtro gaussiano, y obtengo una imagen poco borrosa respecto

a la imagen con grises, Y esta imagen ya no es afectada por un único píxel de ruido con grado significativo como se puede observar en la Figura No. 29.

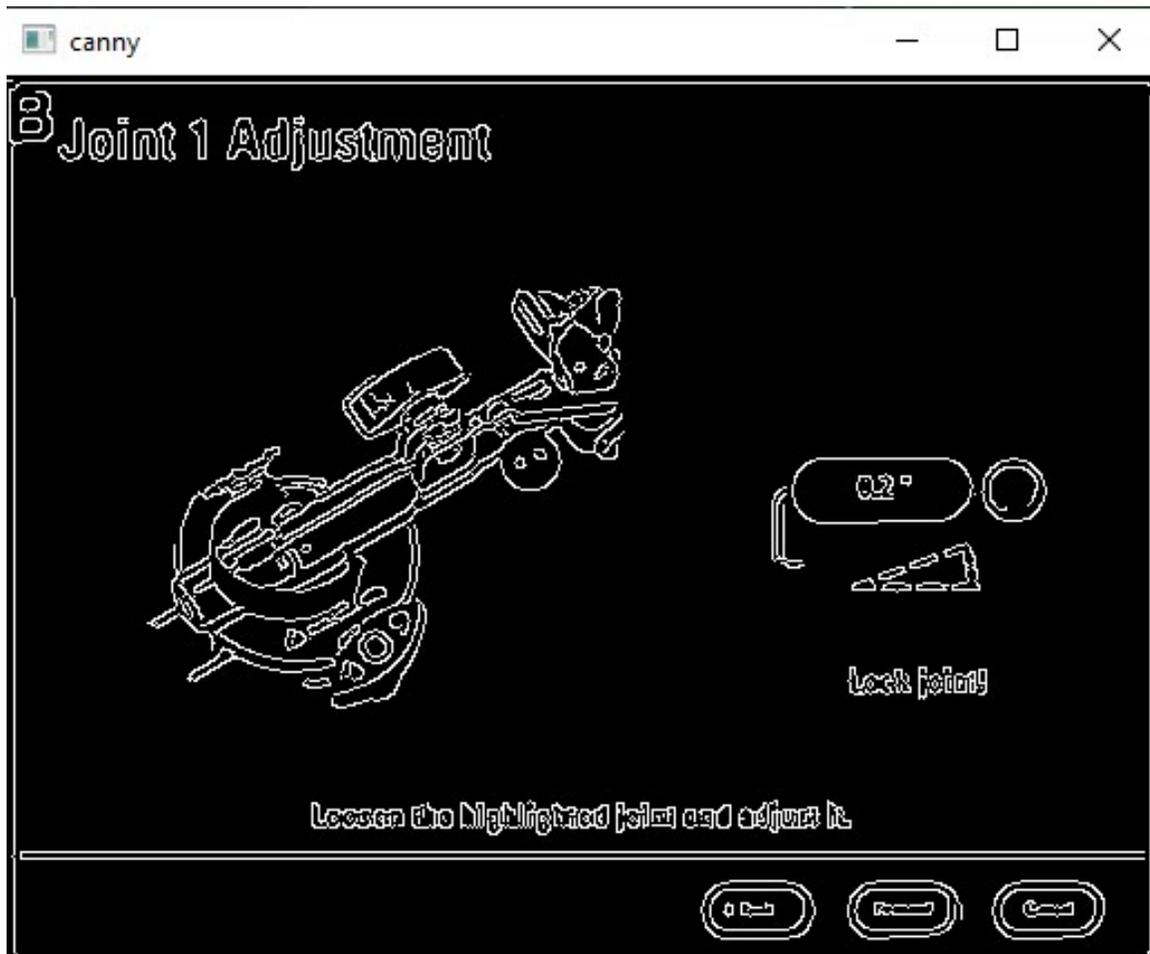


Figura 29: Filtro canny

Con esto se obtienen los bordes, ya que el filtro canny me ayudo a detectar los bordes horizontales, verticales y diagonales. Y estos datos de bordes se obtienen con las derivadas de la gradiente de la Matrix de 5x5 (Figura No. 30) que utiliza el filtro gaussiano.

$$\mathbf{B} = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * \mathbf{A}$$

Figura 30: Matrix gaussiana

Este filtro se enfoca en los pixeles centrales y no en los alrededores que es lo que se busca usar en este programa de automatización para detectar los bordes de interés, (Figura No.31).

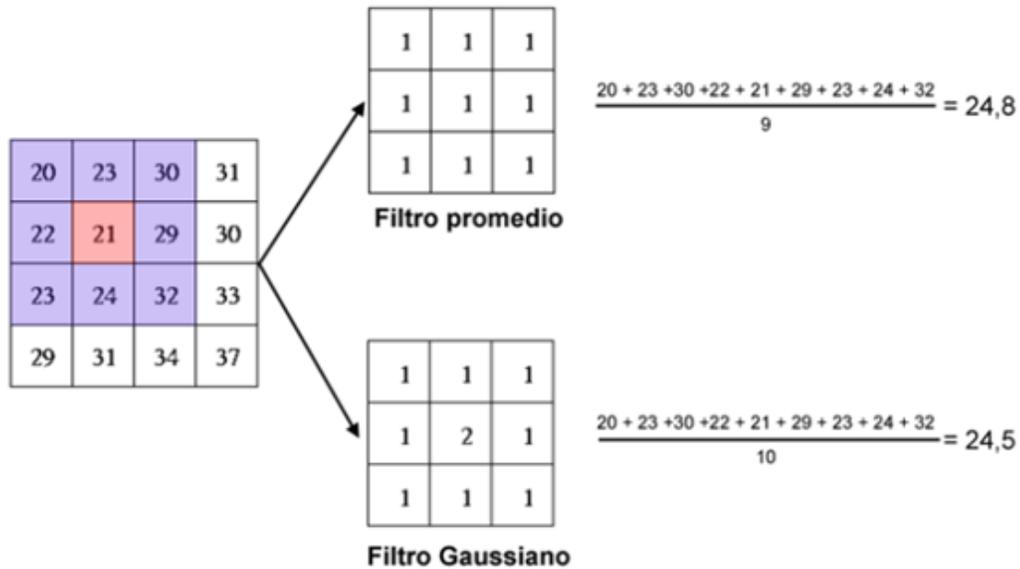


Figura 31: Enfoque de pixeles

Luego de obtener los contornos con el filtro de canny, se agarran estos de la imagen con la función `cv2.findcontours`. Con esta función nos aseguramos de agarrar todas las esquinas ya obtenidas de Canny ya que se tienen los pixeles ya intensificados de las esquinas. Con esto se encuentra una jerarquía con el con contorno mayor y menor encontrado.

Se usan las funciones de `CVRETRLIST` para encontrar los parámetros de los contornos y establece las jerarquías ya antes mencionadas. También se usó la función de `CVCHAIN` el cual considera los puntos de los contornos digamos esto me ayuda a obtener los puntos principales de las figuras que en este caso serían las figuras de mi interés, Yo busco encontrar estos polígonos que encierran donde está el porcentaje de error.

Luego de obtener las áreas les hago un filtro de áreas donde busco el tamaño y el contorno del radio de aspecto, estas dependen de la altura y ancho de los datos ya antes obtenidos, busco que el área no sea menor a 1000 pixeles y mayor 20000 pixeles, con esto aseguro que la imagen, aunque sea afectada por que sea más grande o pequeña siempre encuentre el área que deseo, y luego comparo el radio de aspecto, que siempre va a ser mayor a 2.7 unidades de ancho por una unidad de alto, los demás datos los ignora. No uso otro filtro para identificar más áreas ya que con estos 2 datos ya aseguro que siempre se va a encontrar la Figura No. 32 deseada.



Figura 32: Contorno de interés con canny

8.4. Filtrado de área de interés

Con el procedimiento aplicado en la sección 8.3 se obtienen las coordenadas y el área interés donde se encuentra el error reportado en Brainlab. Se utiliza esta información con la imagen original para aplicarle los mismo filtros que la sección 8.2 y se obtiene la imagen mostrada en la Figura No.33.



Figura 33: Contorno de interés con imagen original

Se usa la función de `cv2.resize` para aumentar el tamaño de la escala de la imagen. La imagen resultante es de 24 bits como en la Figura No.34, pero para usarla en tesseract se necesita en blanco y negro.



Figura 34: Contorno de interés con resize

Se pasa la imagen a escala de grises para poder pasar la imagen a una de 8 bits como en la Figura No.35, para poder obtener una imagen más fácil de controlar.



Figura 35: Contorno en escala de gris con rezise

Se usa un Threshold inverso para que los números estén en blanco y el fondo en negro como en la Figura No. 36. Con esto se soluciona un problema con OCR ya que este no es capaz de distinguir los números de otra forma, ya que en la mayoría de los sistemas de procesamiento de imágenes se procede a obtener datos de pixeles binarios o de 2 niveles para reducir la carga computacional y a la complejidad de los algoritmos. Para esto hay que obtener de una imagen una binarización robusta para poder reducir cualquier error introducido ya sea de la cámara o del entorno en el que se encuentre el sistema.



Figura 36: Contorno de interés en threshold y resize

El problema del método utilizado para poder binarizar la imagen es que las imágenes usadas, o la imágenes obtenidas de HUMANA no cumplen con las condiciones ideales para un procesamiento de imágenes, ya que estas imágenes no son uniformes y usan diferentes colores para los pixeles de los caracteres, lo que significa que al momento de procesar la imagen esta va a tender o a solo blanco o a solo negro, que en nuestro caso es el peor ya que OCR no puede identificar los caracteres dentro del área ya encontrada.

Esta binarización se dificulta a mayor escala si se usa cámara, porque la imagen ya es afectada por variables del entorno, como brillo del entorno, brillo de la computadora, sharpness de la con solo de HUMANA, enfoque de la cámara, etc.

8.4.1. Obtención de datos por inversión de colores

Para resolver el problema de binarización que tenemos donde los caracteres son blancos y los píxeles negros se necesita usar otro sistema de binarización, donde se identificaran los caracteres y se procederán a cambiar todos sus píxeles a negro, y el fondo al igual se identificara y se cambiara a blanco.

Con un filtro de threshold inverso se puede lograr obtener el resultado deseado como se puede observar en la Figura No. 37, pero los resultados con este procedimiento no fueron siempre exitosos pues dependen de la calidad de la imagen.



Figura 37: Valor del error de pantalla en Brainlab

8.4.2. Obtención de datos por binarización promediada

Niblack propuso un esquema de binarización donde el umbral se deriva de las estadísticas de imágenes locales, la media muestral $\mu(x, y)$, y la desviación estándar $\sigma(x, y)$, los cuales están centradas en la ventana W en la ubicación de píxeles (x, y) para que se calcule el umbral T con la fórmula de Umbral de la Figura No. 38 .

$$T_{(x,y)} = \mu_{(x,y)} - k \sigma_{(x,y)}, \quad k = 0.2$$

Figura 38: Ecuación umbral

Como pudimos observar en el procedimiento de canny, este obtiene una imagen donde resalta los valores binarios, Yanowitz y Brucstein introdujeron un umbral que varía en regiones de la imagen para que se adapte espacialmente a las condiciones de iluminación y fondo, como se puede observar en la Figura No. 39. Con el método ya mejorado se obtiene que el fondo hipotéticamente va a tener un valor binarizado de 255 y los píxeles van a tener un valor asignado de 0.

El umbral es calculado con el rango dinámico de desviación estándar, la cual tiene la manera de amplificar la contribución de la desviación estándar de manera adaptativa.

$$T_{(x,y)} = \mu_{(x,y)} \left[1 + k \left(\frac{\sigma_{(x,y)}}{R} - 1 \right) \right]$$

Figura 39: Threshold dinámico

Donde los parámetros R y K se establecen en 128 y 0.5 respectivamente. Este método minimiza el efecto de fondo de ruido, y es más adecuado para imágenes de documentos.

El método nuevo que se va a utilizar usa un componente conectado que se basa en los bordes. la detección de los bordes se realiza individualmente en cada canal de la imagen en color y el mapa de bordes E es obtenido mediante la combinación de las siguientes tres imágenes de borde con la ecuación de:

$$\mathbf{E} = \mathbf{E}_R \vee \mathbf{E}_G \vee \mathbf{E}_B$$

Figura 40: Mapa de bordes

Donde los E son los tres canales de color pertenecientes de RGB y la V denota la operación logica OR.

Con un etiquetado de ocho componentes conectados sigue el borde, paso de la información del cuadro delimitador asociado. A cada componente obtenido se le denota como (EB). Con las suposiciones de que las áreas y las relaciones de los radios de aspecto pertenecientes a EB van a ser utilizadas para filtrar las regiones de texto obvias.

Anteriormente si ya se usó un aspecto de radio para poder delimitar el área para poder enfocar solo el ovalo donde se encuentra el porcentaje de error de los motores. Ahora este nuevo radio de aspecto va a ser entre 0.1 y 10 para limitar regiones grandes. El tamaño de EB debe de ser mas grande 15 pero mas pequeño que 1/5 parte de la imagen para que pueda ser procesada y no agarrar información no deseada.

Como la detección de bordes captura la parte interior y exterior de los limites de los caracteres, es posible que EB ya este contenido una o varias veces, como la letra O, este da lugar a dos compontes. Uno que esta denotado en el límite interno y el otro en el limite externo. Si Un EB contiene uno o más EBs encerrados dentro del estos pueden ser ignorados ya que ya fueron encerrados por los limites interiores. Con esto solo se deja el EB que están dentro del carácter, y los EBs del exterior son removidos ya que no representan al carácter , y estos EBs son filtrados de la siguiente manera como se puede observar en la Figura No. 41.

```

if (Nint < 3)
  {Reject EBint, Accept EBout}
else
  {Reject EBout, Accept EBint}

```

Figura 41: Filtrado de borde

Donde Eb del interior denota a el EB que está en N interior son los EBs totales que hay dentro de la figura: estas restricciones remueven por complemento componentes que no sean parte de algún texto, los EBs que están filtrados son considerados para la binarizacion como en la Figura No. 42.



Figura 42: Filtrado de caracteres

Para cada EB, se calcula la intensidad de primer plano y la intensidad del fondo, y el umbral se calcula individualmente. La intensidad del primer plano se calcula con niveles de grises para cada pixel donde corresponde a un limite del carácter como se observa en la Figura No. 43 .

$$F_{EB} = \frac{1}{N_E} \sum_{(x,y) \in \mathbf{E}} \mathbf{I}(x,y)$$

Figura 43: Intensidad de primer plano

Donde E representa las esquinas del carácter, I(x, y) representa la intensidad de los pixeles de (x,y) y NE es el numero de esquinas que componente el carácter.

Estos componentes se pueden verificar en la Figura No. 44.

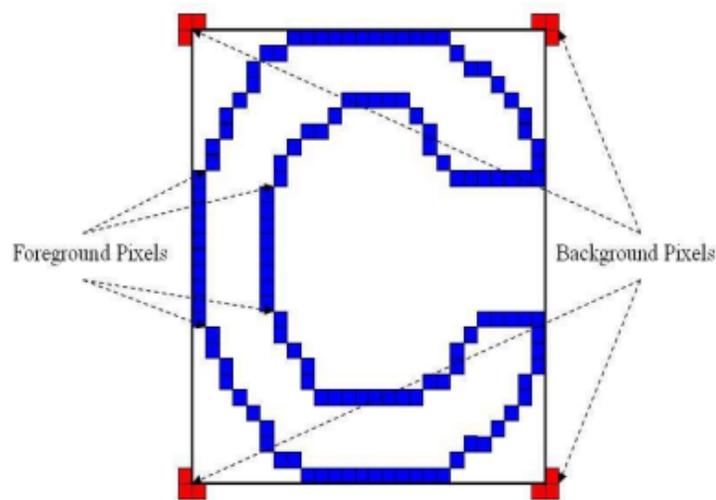


Figura 44: Caracter encontrado

Para obtener la intensidad del fondo se consideran 3 pixeles. Cada uno en la periferia de las esquinas del cuadro delimitador de la siguiente manera (Figura No.45).

$$\mathbf{B} = \{\mathbf{I}(x - 1, y - 1), \mathbf{I}(x - 1, y), \mathbf{I}(x, y - 1), \\ \mathbf{I}(x + w + 1, y - 1), \mathbf{I}(x + w, y - 1), \mathbf{I}(x + w + 1, y), \\ \mathbf{I}(x - 1, y + h + 1), \mathbf{I}(x - 1, y + h), \mathbf{I}(x, y + h + 1), \\ \mathbf{I}(x + w + 1, y + h + 1), \mathbf{I}(x + w, y + h + 1), \mathbf{I}(x + \\ w + 1, y + h)\}$$

Figura 45: Periferias de las esquinas

Donde (x, y) representan las coordenadas del lado superior izquierdo de las esquinas de cada cuadro delimitador del componente de cada esquina, donde h es la altura y w el ancho.

Para casos donde la imagen es lineal la intensidad media debería de ser igual que los pixeles del primer plano, pero esto puede verse afectado si la imagen no es lineal, ya que son los cuadros delimitados pueden presentar algún tipo de interferencia en el fondo de estimación.

La intensidad de fondo media para los componentes no lineales puede verse afectado, pero la mediana no es afectada, por lo tanto, la intensidad del fondo local se puede estimar de manera más confiable, considerando la media de la intensidad de 12 pixeles en lugar de la intensidad media. Como se puede observar en la Figura No. 46.

$$B_{EB} = \text{median}(\mathbf{B})$$

Figura 46: Intensidad media

Asumiendo que cada carácter ya es de color uniforme, se puede hacer la binarización de cada esquina de cada componente usando la intensidad del primer plano de threshold. Dependiendo si la intensidad es alta o baja que el fondo, cada salida binarizada BW es invertida y el texto queda negro, y el primer plano queda en blanco. (Se puede verificar en la Figura No. 47).

$$\text{If } F_{EB} < B_{EB}, \mathbf{BW}_{EB}(x, y) = \begin{cases} 1, & \mathbf{I}(x, y) \geq F_{EB} \\ 0, & \mathbf{I}(x, y) < F_{EB} \end{cases}$$

$$\text{If } F_{EB} > B_{EB}, \mathbf{BW}_{EB}(x, y) = \begin{cases} 0, & \mathbf{I}(x, y) \geq F_{EB} \\ 1, & \mathbf{I}(x, y) < F_{EB} \end{cases}$$

Figura 47: Binarización

El resultado final del proceso se puede verificar en la Figura No. 48) [23]



Figura 48: Resultado de la binarización

8.5. Resultados de filtrados con diferentes cámaras

El sistema anterior se diseñó para poder ser usado con imágenes descargadas en el sistema, pero el sistema que se planificó es que estas imágenes provinieran directamente de una cámara web. Esta cámara deberá de estar instalada en el centro de HUMANA para poder obtener la información en directo de la pantalla de Brainlab.

8.5.1. Resultados con GoPro Hero 8

La cámara utilizada para poder automatizar el proceso fue una cámara GoPro Hero 8 como en la Figura No. 49, esta GoPro tiene bastantes funciones, las más importantes que tenían que usar eran la de webcam y la de fotografías. Se decidió usar la de fotografía ya que las imágenes tienen más resolución que la imagen de la configuración de Webcam.

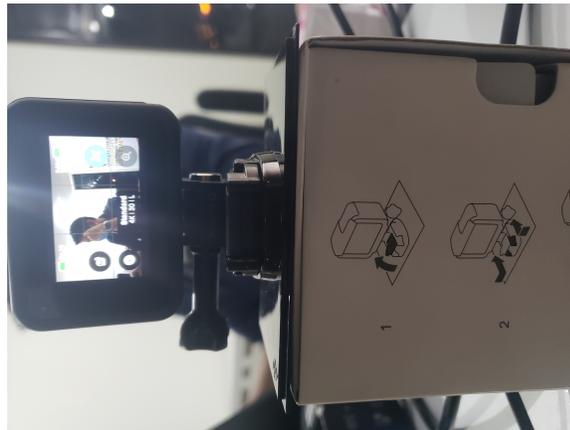


Figura 49: Cámara GoPro

- El shutter en automático. (default)
- El balance de blancos en automático (default)
- ISO Max 3200.(default)
- ISO Min 100 (default)

- Sharpness (default)
- La salida de imagen en Standard
- Lente linear (default)
- Zoom en 1 (default)
- Lente linear (default)
- La distancia entre la cámara y la pantalla para que el ORC no falle, depende directamente de lo que ve la cámara, y en este caso la cámara tiene que encuadrar a toda la pantalla.

Para que se puedan obtener resultados consistentes con esta cámara se usaron la mayoría de las configuraciones en default. Asegurando así que los resultados fueran casi similares.

Al principio se obtiene la imagen normal de 24 bits como se puede observar en la Figura No. 50.

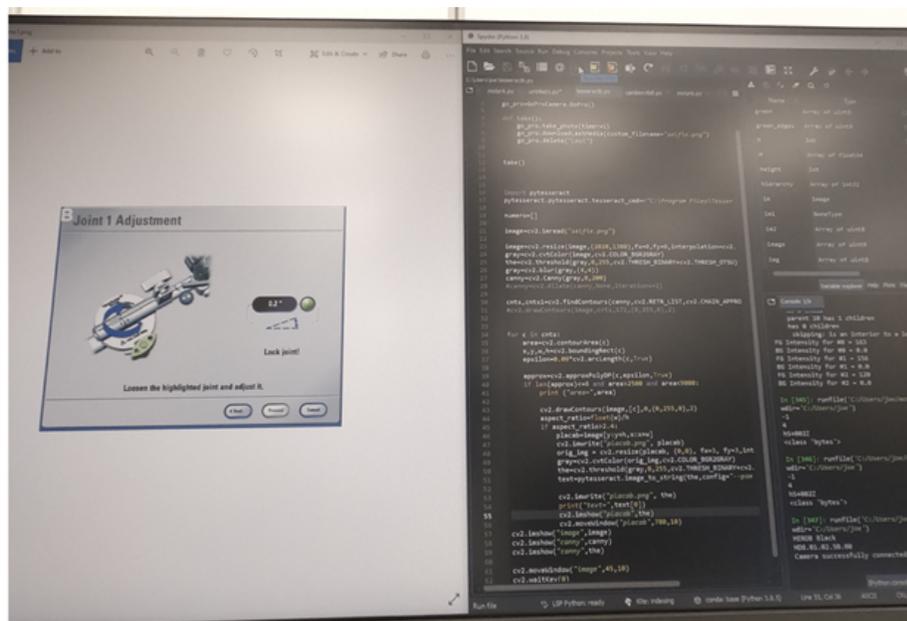


Figura 50: Imagen con filtro inicial

Luego se uso un filtro con la imagen obtenida y pasarla a escala de grises con 8 bits como se puede observar en la Figura No. 51

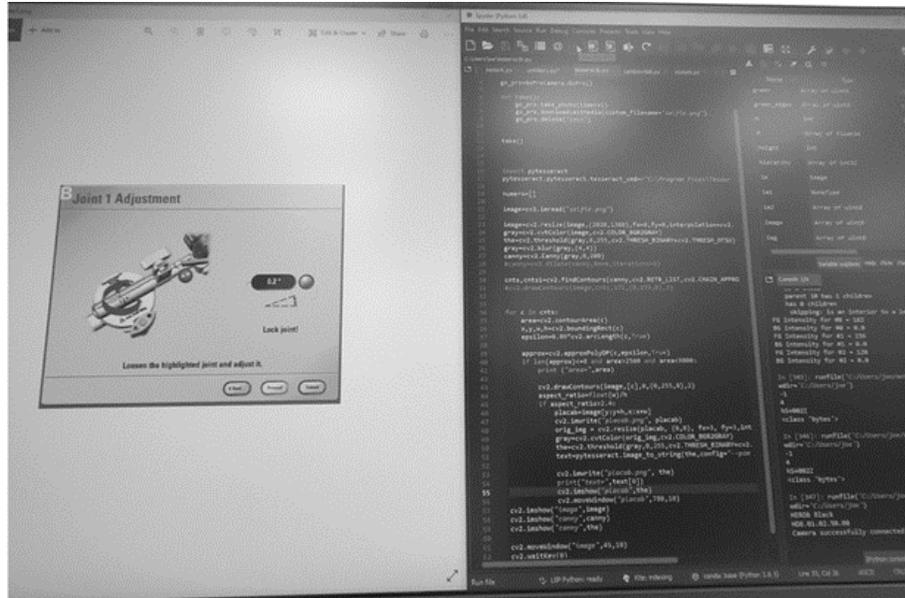


Figura 51: Imagen con filtro grises

Con la imagen en escala de grises se prosigue a pasar la imagen a binario en blanco y negro con un filtro de Threshold. La imagen se observa en la Figura No. 52.

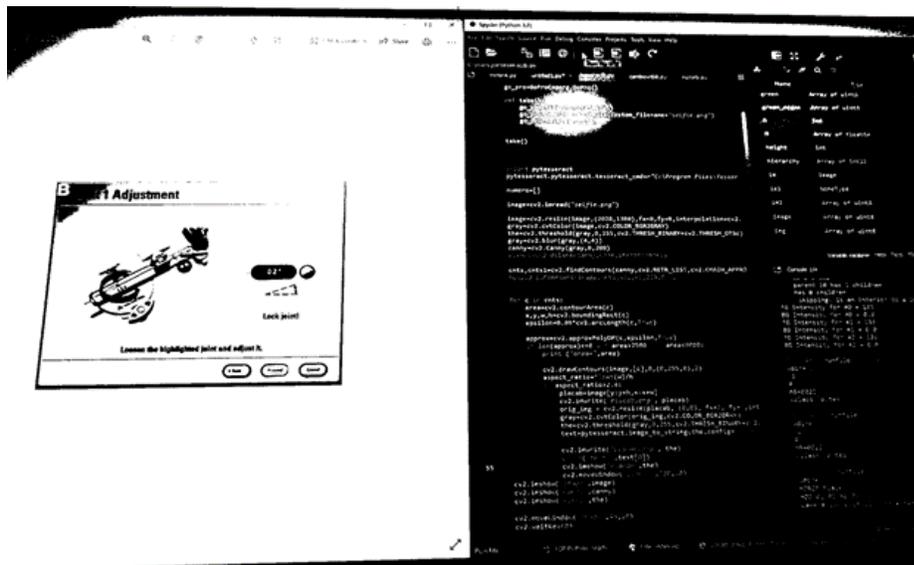


Figura 52: Imagen con filtro Threshold

Con esto se prosigue a obtener el área de interés con un filtro de canny y usando las mismas configuraciones que en la sección 8.3, se puede observar el resultado en la Figura No.53.

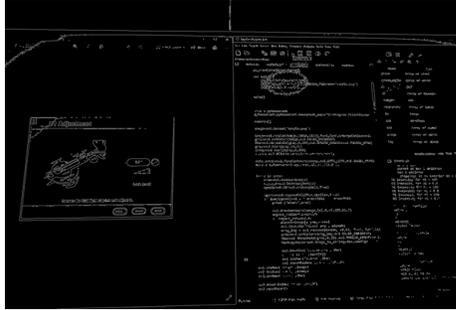


Figura 53: Imagen con filtro canny

Con los parámetros usados en el filtro canny se obtiene el área de interés y se usan estos mismos, pero con la imagen original para poder hacer un procesamiento más exhaustivo, se puede observar el resultado en la Figura No.54.



Figura 54: Imagen con área obtenida

Se usa la función resize de la imagen original para aumentar la escala de la imagen, pero solo con los pixeles donde está el área de interés ignorando lo demás como se puede verificar en la Figura No. 55.



Figura 55: Imagen con área agrandada

Con la imagen en escala de grises y con resize se prosigue a pasar la imagen a binario en blanco y negro con un filtro de Threshold. La imagen se observa en la Figura No. 56.



Figura 56: Imagen con área agrandada en Threshold

Luego se usa un filtrado de Threshold especial para obtener del área de interés claro, se puede observar en la Figura No.57.



Figura 57: Imagen con área con limpieza y binarización invertida

8.5.2. Camara logitech 1080p HD

Para que se puedan obtener resultados consistentes con esta cámara se usaron la mayoría de las configuraciones en default. Asegurando así que los resultados fueran casi similares. El único ajuste que fue necesario sería que la distancia entre la cámara y la pantalla es de 12 a 14 cm enfocando solamente la parte de la pantalla donde están los ajustes de las juntas.

Al principio se obtiene la imagen normal de 24 bits como se puede observar en la Figura No. 58.



Figura 58: Imagen con filtro inicial

Luego se uso un filtro con la imagen obtenida y pasarla a escala de grises con 8 bits como se puede observar en la Figura No. 59

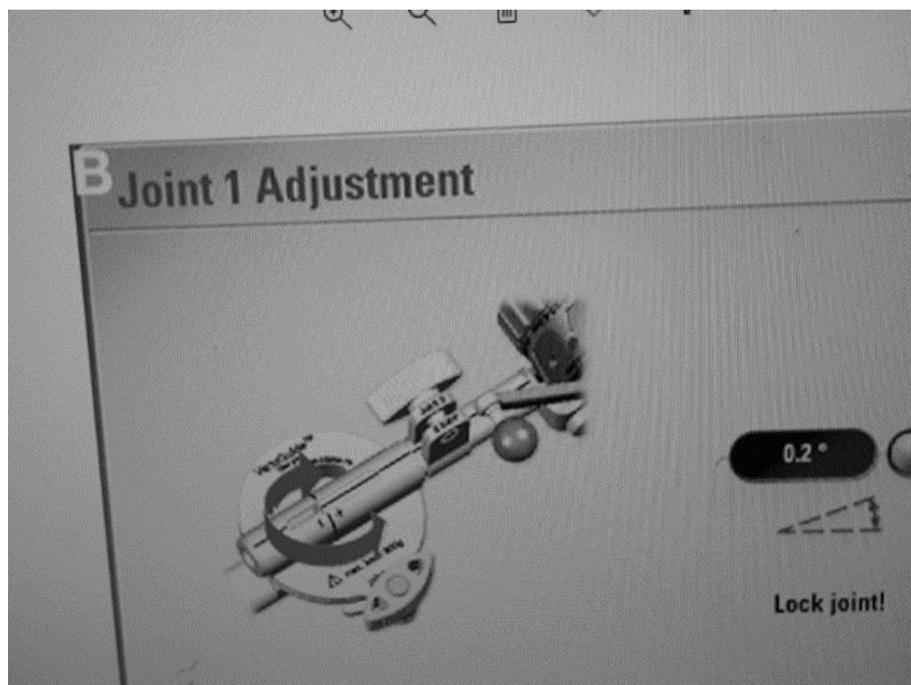


Figura 59: Imagen con filtro grises

Con la imagen en escala de grises se prosigue a pasar la imagen a binario en blanco y negro con un filtro de Threshold. La imagen se observa en la Figura No. 60.

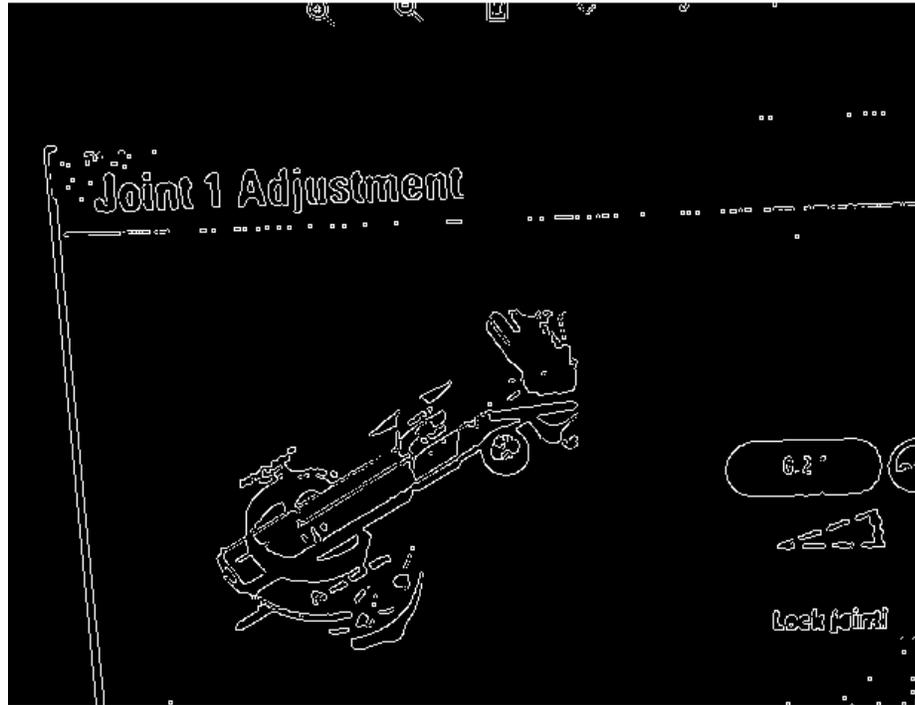


Figura 60: Imagen con filtro Threshold

Con esto se prosigue a obtener el área de interés con un filtro de canny y usando las mismas configuraciones que en la sección 8.3, se puede observar el resultado en la Figura No.61.

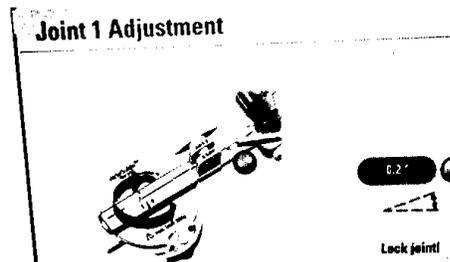


Figura 61: Imagen con filtro canny

Con los parámetros usados en el filtro canny se obtiene el área de interés y se usan estos mismos, pero con la imagen original para poder hacer un procesamiento más exhaustivo, se puede observar el resultado en la Figura No.62.



Figura 62: Imagen con área obtenida

Se usa la función `resize` de la imagen original para aumentar la escala de la imagen, pero solo con los píxeles donde está el área de interés ignorando lo demás como se puede verificar en la Figura No. 63.



Figura 63: Imagen con área agrandada

Con la imagen en escala de grises y `resize` se prosigue a pasar la imagen a binario en blanco y negro con un filtro de `Threshold`. La imagen se observa en la Figura No. 64.



Figura 64: Imagen con área agrandada en `Threshold`

Luego se usa un filtrado de `Threshold` especial para obtener del área de interés claro, se puede observar en la Figura No.65.



Figura 65: Imagen con área con limpieza y binarización invertida

8.6. Filtración de datos y envío de datos por serial

Al momento de obtener la imagen ya filtrada, donde los caracteres son de color negro, y el fondo blanco, se procede al uso de tesseract para poder identificar las variables.

Se usa la función de *pytesseract.image_tostring* con la configuración de psm 11 que sirve para encontrar la mayor cantidad de texto posible, para no perder datos como algún punto o algún signo, como se puede observar en la Figura No. 66.

text	str	1	0.2°
texta	str	1	02°
texta3	str	1	Joint 1 Adjustment
textb	str	1	02°
textc	str	1	Wed
textraf	str	1	+002
textrafb	str	1	fS+002Z
textrafb1	bytes	1	fS+002Z

Figura 66: Valor del error de pantalla en Brainlab

Las consideraciones usadas para el filtrado de datos son las siguientes (se puede visualizar el resultado en el gráfico en la Figura No. 67) :

- Se necesita poder enviar datos en paquetes de bytes por medio de comunicación Serial, con tesseract se obtienen strings así que hay que modificar la información.
- El primero debe de representar en minúscula el valor del motor a mover.
- El segundo debe de ser una S en mayúscula representando el inicio de la comunicación serial.

- El tercero el signo si es positivo o negativo.
- El cuarto, quinto y sexto tienen que ser el valor que indica la pantalla de HUMANA.
- El séptimo y último valor tiene que ser una Z en mayúscula para indicar el final del string.
- Enviar los datos en paquetes de bytes y no strings.

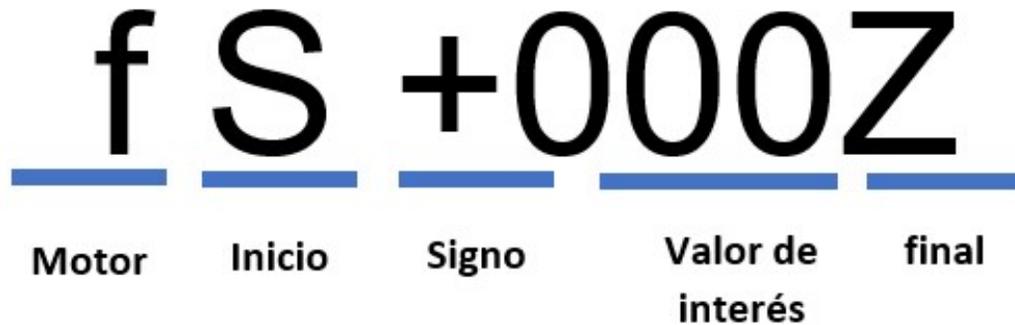


Figura 67: Datos que espera el sistema de control principal

Luego de esto se busca encontrar algún tipo de dato incorrecto por lo tanto se obtiene el valor de longitud de caracteres que se obtienen de tesseract y como se que el texto no puede ser mayor a 7 caracteres, se hace un filtrado hasta 4 caracteres y si encuentra un dato que no pertenece al porcentaje de error de algún motor, este solamente se ignora fácilmente.

Ya que HUMANA no siempre enseña 3 valores, la unidad y decimal se debe de verificar si hay decena o no, además de verificar si es positivo o negativo el ángulo.

Luego se verifica si tiene signo o no, si tiene signo significa que es negativo ya que la pantalla de HUMANA solo puede dar signos negativos y no positivos. Si el dato pasa por esta identificación se verifica la posición del punto. Ya que si el punto esta en la tercera posición esto significa que solo hay 2 valores entonces se le debe de agregar un 0 a la cuarta posición de la información que se va a enviar. Entonces se agregan los datos del porcentaje de error después de la cuarta posición y se adjunta el motor al que corresponde.

Si el dato verificado en la secuencia no tiene signo significa que es positivo, entonces a la tercera posición se le agrega un + seguido de los otros datos. Y si el punto que se obtuvo de tesseract está en la segunda posición significa que no hay decenas, por lo que se le agrega un 0 en la cuarta posición. Entonces se agregan los datos del porcentaje de error después de la cuarta posición y se adjunta el motor al que corresponde en la primera posición del array, además se agrega en la segunda posición de esta un letra S en Mayúscula que representa el inicio de la comunicación serial en el sistema de control principal y luego se agregan los 3 valores del error. El array obtenido debe de finalizar con una Z que representa el final del dato.

Por medio de protocolo serial se envían los datos a un sistema de control principal. Los datos se convierten de strings a BYTES para que no haya problemas con la lectura de este. El resultado de la verificación se puede observar en la Figura No. 68.

```
10
v1 0
0.2°

i
v2 0
0.2°

v3 0
0.2°

textra2= +02

textra3= +002
+002
fS+002Z
b'fS+002Z'
```

Figura 68: Resultado de verificación de datos

Implementación física del anclaje de cámara

La parte mecánica consiste en un brazo que iría en la base de la pantalla en HUMANA, esta cámara no debería de poder configurarse, ya que el doctor debería de poder usar la cámara ni ajustar ningún detalle y usarla directamente. Pero por problemas por la pandemia no podíamos ir a HUMANA con frecuencia por lo que no se obtuvo retroalimentación para poder configurar la distancia de la cámara o obtener cálculos para poder hacer una base estática.

9.1. Primer prototipo

Por la opción más razonable es hacer un brazo que pueda ajustar distancia de la cámara a la pantalla y el ángulo de giro a la pantalla. Este brazo debería de ser rígido y fácil de ajustar ya que se usaría principalmente para hacer pruebas en HUMANA, y con estos datos poder hacer un brazo rígido que no necesite configurarse para que solo sea de poner y quitar. Teniendo esto en cuenta se diseño un brazo capaz de poder ajustar distancia y ángulo.

Para poder cambiar la configuración de distancia se decidió hacer cilindros de distintos tamaños y con diámetro diferentes para que se acoplaran con facilidad, estos cilindros estarían impresos en PLA y se imprimirían en impresora 3D. El resultado de la verificación se puede observar en la Figura No. 69.



Figura 69: Cilindros de PLA

Para el mecanismo que gira se tenía pensado hacer cilindros que pudieran girar con un pin como se puede observar en la Figura No. 70, este diseño permitiría al brazo mecánico poder girar y ajustar el ángulo para ajustar la cámara. Se tenía planeado diseñar con material PLA.



Figura 70: Sistema giratorio

El resultado final fue el siguiente (el resultado de la simulación se puede verificar en la Figura No. 71 :

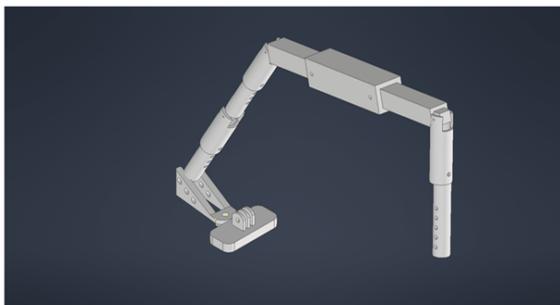


Figura 71: Prototipo inicial

9.2. Segundo prototipo

El diseño anterior fue rechazado ya que no aseguraba que el mecanismo no girara por lo que decidió ir por una opción más rígida que en este caso era una base giratoria que se podría ajustar al ángulo. Este diseño podría permitir hacer giros grandes, y con pines en ambos lados se podrían hacer que el sistema giratorio fuera rígido. (El resultado se puede verificar en la Figura No. 72).



Figura 72: Sistema giratorio

9.3. Sistema giratorio

Luego de pensar en los detalles del brazo, en lugar de imprimir los cilindros que darían la corrección de distancia se pensó en cambiarlo por un material resistente y liviano como el de acero galvanizado, este material se consiguió en 2 tamaños de diámetro de 0.5 pulgadas y de 0.75 pulgadas. Con estos 2 tamaños de diámetro se podría hacer la configuración necesaria para poder cambiar la distancia con facilidad. (El resultado de la simulación se puede verificar en la Figura No. 73).



Figura 73: Cilindros de acero galvanizado

Para asegurarse que el sistema giratorio fuera rígido se decidió usar pines en una sola pieza, para que esta se pueda acoplar a la otra y creara la rigidez deseada para este sistema. (El resultado de la simulación se puede verificar en la Figura No. 74).



Figura 74: Sistema giratorio con pines

La base del sistema iría en un computador para poder simular como iría en el sistema en la clínica HUMANA, para esta base se decidió instalar directamente en la pantalla ya que esta tenía agujeros para poder tornillos. Esta iría a un acople cilíndrico para poder ajustar con el cilindro de acero galvanizado.

Para ajustar el ángulo de la cámara se decidió usar un acople giratorio en el cual podría hacer girara la cámara media vez no estaría ajustado. Este acople iría directamente conectado con un tubo cilíndrico para ajustar la altura de la cámara. (El resultado de la simulación se puede verificar en la Figura No. 75).



Figura 75: Pieza para hacer girar la camará

El sistema final quedó de tal manera que se acopla con facilidad al computador, el acople angular cumplió con las especificaciones que se querían ya que es muy rígido y no gira al menos que lo desacople. Con esto aseguro que la cámara no gire hasta el momento que el

usuario decida hacerlo. La configuración de distancias también cumplió con las especificaciones ya que podría configurar la distancia con facilidad, y podía acoplarlo con un solo tornillo de 6 mm.

El resultado con la Gopro fue el siguiente y se puede observar en la Figura No. 76:

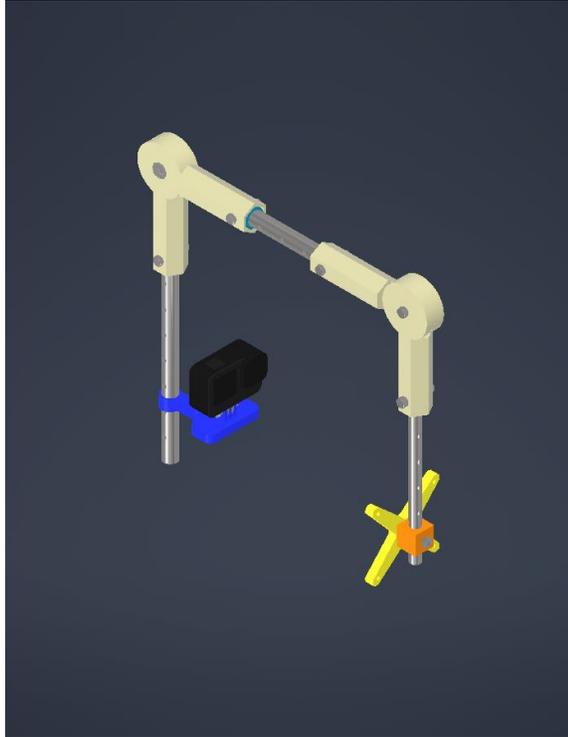


Figura 76: Agarrador de cámara web simulación con GoPro

El resultado con la webcam logitech fue el siguiente y se puede observar en la Figura No. 77:

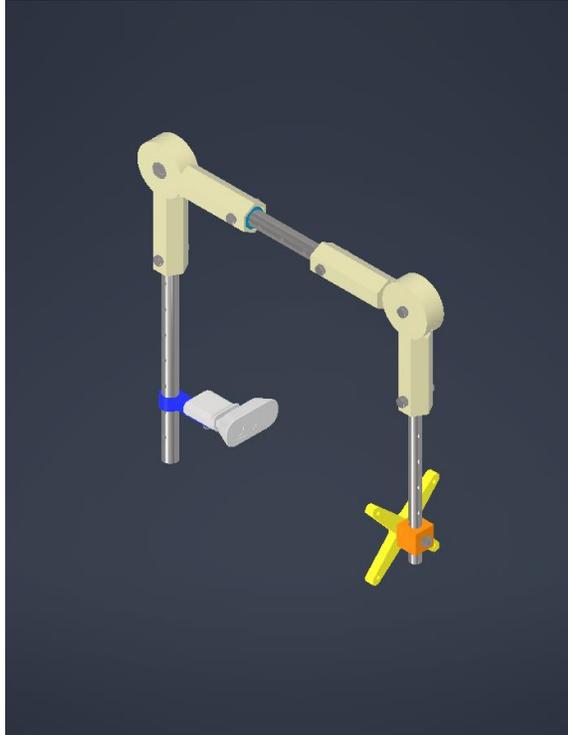


Figura 77: Agarrador de cámara web simulación con Logitech

El resultado final ya contruido fue el siguiente y se puede observar en la Figura No. 78:



Figura 78: Brazo de cámara web con Logitech

9.3.1. Simulación de cargas puntuales

El prototipo físico soportó su propio peso, y eso se puede observar en el análisis de esfuerzos realizado en inventor, justificando que la estructura necesitaba ser mejorada. El análisis se llevo a cabo utilizando la fuerza de gravedad ubicada en centro de gravedad, para comparar el esfuerzo obtenido con el máximo del material. La configuración del brazo que

asegura donde se puede obtener el máximo esfuerzo es cuando está completamente extendido de forma horizontal.

La gravedad está dada desde la parte del centro de la estructura, para comparar los esfuerzos de máximos y ver en qué punto puede fallar. Para simular las cámaras se utilizó un plástico con el que la pieza tuviese un peso similar al de la vida real, siendo este 18gr cada uno. El resultado se puede observar en la Figura No. 79 y Figura No. 80:

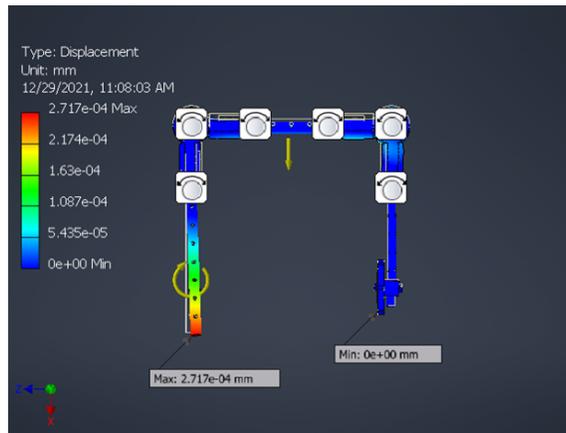


Figura 79: Esfuerzos de Von Mises

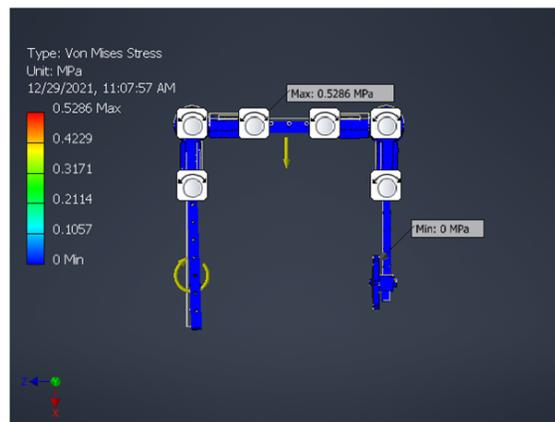


Figura 80: Desplazamiento del material

Si la cámara GoPro o la cámara Logitech están configuradas con los pasos que se detallaron se puede concluir que:

- El sistema OCR permitió encontrar las variables deseables que en este caso eran los porcentajes de error del sistema de varioguide de Brainlab, cabe destacar que el procesamiento de imágenes que utiliza el software no necesita que la pantalla donde se encuentran los porcentajes de error este en un mismo lugar para poder obtener los datos, lo que ayuda a el sistema sea fácil de usar.
- El software diseñado obtiene las variables de Varioguide sin necesidad de estar pulsando botones, lo que permite que el sistema sea automático, pero se agregaron botones al sistema solo para poder observar y garantizar que los datos obtenidos son correctos.
- Las ecuaciones utilizadas para hacer el Thresholding de las imágenes funcionan en cualquier imagen ya que requieren un cambio matricial de los pixeles por lo que cualquier imagen puede ser procesada.
- Los resultados obtenidos por el uso de OCR muestran que este sistema sí es confiable ya que los valores que se obtienen en el programa son idénticos a los de las imágenes.

- Las librerías utilizadas en Python tienen que ser verificadas mediante el uso de programas básicos, ya que pueden presentar errores de diferentes procesos.
- Para el procesamiento de las imágenes se recomienda obtener todos los píxeles en blanco y negro pero que no se mezclen estos colores entre letras y fondos, que OCR solo puede detectar comúnmente dos tonos para diferenciar los caracteres.
- Ir a al centro de HUMANA a hacer pruebas de distancias para la cámara a comprobar funcionamiento y poder diseñar una base fija que sea más intuitiva que la que se diseñó.
- Mejorar el diseño del brazo que se diseñó para HUMANA ya que por falta de pruebas no se pudo realizar uno más preciso.
- Mejorar la aplicación diseñada ya que no se pudieron actualizar las variables encontradas por motivos de estructura del software.
- Hacer pruebas con los cuatro motores, ya que por falta de estos componentes no se pudieron probar todas juntas.

-
- [1] F. Xu, H. Jin, X. Yang y col., “Improved accuracy using a modified registration method of ROSA in deep brain stimulation surgery,” en_US, *Neurosurgical Focus*, vol. 45, n.º 2, E18, ago. de 2018, Publisher: American Association of Neurological Surgeons Section: Neurosurgical Focus, ISSN: 1092-0684. DOI: 10.3171/2018.4.FOCUS1815. dirección: <https://thejns.org/focus/view/journals/neurosurg-focus/45/2/article-pE18.xml> (visitado 14-06-2021).
- [2] S. Pandya, J. W. Motkoski, C. Serrano-Almeida, A. D. Greer, I. Latour y G. R. Sutherland, “Advancing neurosurgery with image-guided robotics: Technical note,” en_US, *Journal of Neurosurgery*, vol. 111, n.º 6, págs. 1141-1149, dic. de 2009, Publisher: American Association of Neurological Surgeons Section: Journal of Neurosurgery, ISSN: 1933-0693, 0022-3085. DOI: 10.3171/2009.2.JNS081334. dirección: <https://thejns.org/view/journals/j-neurosurg/111/6/article-p1141.xml> (visitado 14-04-2021).
- [3] E. Kunze, J. Meixensberger, M. Janka, N. Sörensen y K. Roosen, “Decompressive Craniectomy in Patients with Uncontrollable Intracranial Hypertension,” en, *Intracranial Pressure and Neuromonitoring in Brain Injury*, págs. 16-18, 1998, Publisher: Springer, Vienna. DOI: 10.1007/978-3-7091-6475-4_5. dirección: https://link.springer.com/chapter/10.1007/978-3-7091-6475-4_5 (visitado 14-06-2021).
- [4] D. M. Fountain, A. Bryant, D. G. Barone y col., “Intraoperative imaging technology to maximise extent of resection for glioma: a network meta-analysis,” en, *Cochrane Database of Systematic Reviews*, Cochrane Gynaecological, Neuro-oncology and Orphan Cancer Group, ed., ene. de 2021, ISSN: 14651858. DOI: 10.1002/14651858.CD013630.pub2. dirección: <http://doi.wiley.com/10.1002/14651858.CD013630.pub2> (visitado 14-05-2021).
- [5] A. Omesaña y M. González, “Evaluación Neuropsicológica en la enfermedad de Alzheimer: Memoria Episódica y Semántica,” es, *Cuadernos de neuropsicología*, vol. 3, n.º 2, págs. 199-223, dic. de 2009, Publisher: Neuropsicologia.cl, ISSN: 0718-4123. dirección: http://pepsic.bvsalud.org/scielo.php?script=sci_abstract&pid=S0718-41232009000200006&lng=pt&nrm=iso&tlng=es (visitado 14-06-2021).

- [6] J. A. Ledesma, *La imagen social de las personas con discapacidad. Estudios en homenaje a José Julián Barriga Bravo*, es. CERMI, 2008, Google-Books-ID: EivYreIjoNsC, ISBN: 978-84-96889-32-3.
- [7] Y. LeCun, Y. Bengio y G. Hinton, “Deep learning,” en, *Nature*, vol. 521, n.º 7553, págs. 436-444, mayo de 2015, Bandiera_abtest: a Cg_type: Nature Research Journals Number: 7553 Primary_atype: Reviews Publisher: Nature Publishing Group, ISSN: 1476-4687. DOI: 10.1038/nature14539. dirección: <https://www.nature.com/articles/nature14539> (visitado 15-04-2021).
- [8] *Referencia del Lenguaje Python — documentación de Python - 3.9.6*. dirección: <https://docs.python.org/es/3/reference/index.html> (visitado 15-07-2021).
- [9] G. d. l. B. N. D. G. t. d.-G. g. d. d. dice, *OCR, ICR y OMR. ¿Qué son y para qué sirven?* es-ES, nov. de 2009. dirección: <https://gupost.wordpress.com/2009/11/09/ocr-icr-y-omr-%c2%bfque-son-y-para-que-sirven/> (visitado 15-04-2021).
- [10] *[Tutorial] OCR in Python with Tesseract, OpenCV and Pytesseract*, en, jun. de 2021. dirección: <https://nanonets.com/blog/ocr-with-tesseract/> (visitado 15-03-2021).
- [11] *MATLAB for Students - MATLAB & Simulink*. dirección: https://la.mathworks.com/products/matlab/student.html?s_tid=acport_sv_sp_cta1 (visitado 15-04-2021).
- [12] *Stefan Vilsmeier—Founder and CEO Brainlab*, en-US. dirección: <https://www.brainlab.com/contact-us/journalists/management-board-stefan-vilsmeier/> (visitado 15-03-2021).
- [13] “DARPA’s new 1.8-gigapixel camera is a super high-resolution eye in the sky,” New Atlas. Section: Military. (12 de feb. de 2013), dirección: <https://newatlas.com/argus-is-darpa-gigapixel-camers/26078/> (visitado 03-01-2022).
- [14] J. C. López. “Qué hay que mirar al comprar una cámara fotográfica para asegurarnos de que va a tener las tecnologías de los próximos años,” Xataka. (2 de mayo de 2020), dirección: <https://www.xataka.com/fotografia-y-video/qhay-que-mirar-al-comprar-camara-fotografica-para-asegurarnos-que-va-tener-tecnologias-proximos-anos-1ue> (visitado 03-01-2022).
- [15] “Las ópticas CCD o dispositivos de carga acoplados.” Section: Blog. (), dirección: <https://galernaestudio.com/optica-ccd-dispositivo-de-carga-acoplado/> (visitado 03-01-2022).
- [16] “Filtro de Bayer,” HiSoUR Arte Cultura Historia. (18 de mar. de 2018), dirección: <https://www.hisour.com/es/bayer-filter-24546/> (visitado 03-01-2022).
- [17] “Kueppers teoria del color: Homepage.” (9 de feb. de 2008), dirección: <https://web.archive.org/web/20080209023216/http://www.ipsi.fraunhofer.de/Kueppersfarbe/es/index.html> (visitado 03-01-2022).
- [18] “Fotografía Digital - Concepto, historia y características.” Section: Arte. (), dirección: <https://concepto.de/fotografia-digital/> (visitado 03-01-2022).
- [19] “Fundamentos y evolución de la multimedia » Zoom óptico y Zoom digital.” (), dirección: <http://multimedia.uoc.edu/blogs/fem/es/zoom-optico-y-zoom-digital/> (visitado 03-01-2022).
- [20] “Image Formation by Lenses.” (), dirección: <http://hyperphysics.phy-astr.gsu.edu/hbasees/geoopt/lensdet.html> (visitado 03-01-2022).

- [21] “La Sensibilidad ISO | PDF | Velocidad de la película | Velocidad de obturación.” (), dirección: <https://www.scribd.com/document/374193062/4-La-Sensibilidad-ISO> (visitado 03-01-2022).
- [22] “¿Qué es el color de luz y la temperatura de color? - Blog,” LED Studio. (), dirección: https://www.ledstudio.cl/blog/4_que-es-el-color-de-luz-y-la-temperatura-de-color.html (visitado 03-01-2022).
- [23] J. Funk, *ocr-text-extraction*, original-date: 2012-08-04T14:39:20Z, 1 de ene. de 2022. dirección: https://github.com/jasonlfunk/ocr-text-extraction/blob/2ca74e9faf57caa5b6f5dcfextract_text (visitado 03-01-2022).

13.1. Programación

Link de programación <https://github.com/vaelan98/tesis>

13.2. Protipo sujetador de web cam en HUMANA

El prototipo funcional de Humana se realizó tomando en cuenta los siguientes criterios:

- El brazo debe ser muy facil de acceder o manipular.
- El brazo debe de contar con solo un grado de libertad.
- El prototipo diseñado debe soportar las condiciones propuestas por el diseño del mecanismo físico.

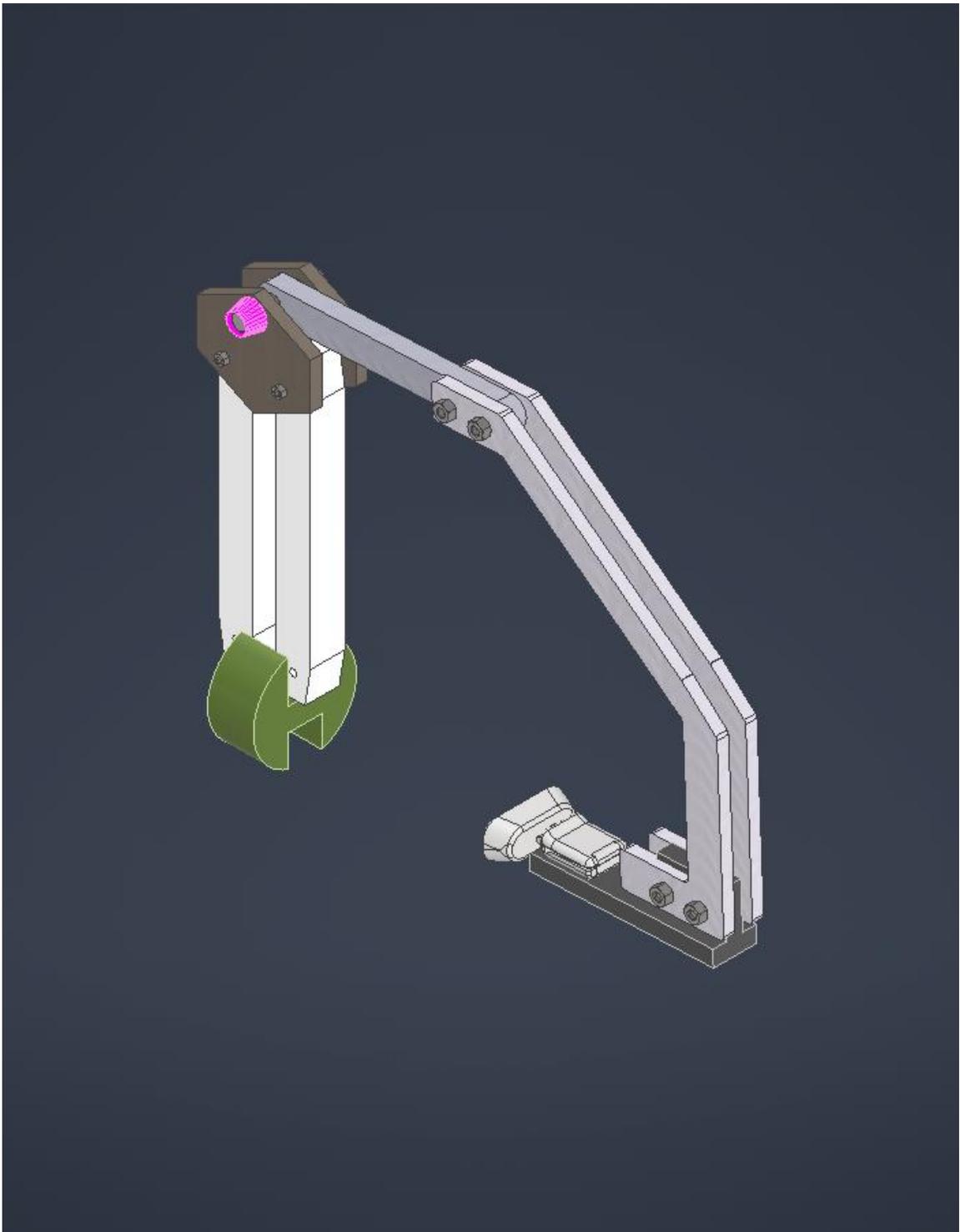


Figura 81: Brazo de HUMANA

