UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO
CARRERA DE INGENIERÍA ELECTRÓNICA

DIAGNÓSTICO DEL ESTADO DE TARJETAS MADRE A TRAVÉS DE INTELIGENCIA ARTIFICIAL Y TERMOGRAFÍA

Trabajo de titulación previo a la obtención del
Título de Ingeniero Electrónico

AUTOR: Stalin Roberto Villalba Armendáriz
TUTOR: Víctor Vinicio Tapia Calvopiña

Quito-Ecuador
2023

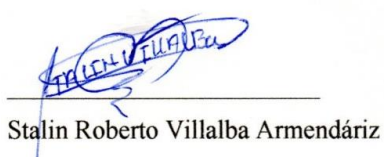
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Stalin Roberto Villalba Armendáriz con el documento N° 1725359606 manifiesto que:

soy autor y responsable del presente trabajo, y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 06 de marzo del año 2023

Atentamente,



Stalin Roberto Villalba Armendáriz

1725359606

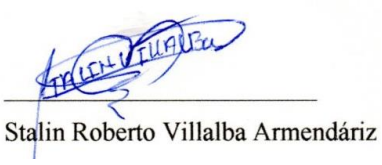
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Yo, Stalin Roberto Villalba Armendáriz con el documento N° 1725359606 expreso con mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales de virtud de que soy el autor del Proyecto Técnico: "Diagnóstico del estado de tarjetas madre a través de inteligencia artificial y termografía", el cual ha sido desarrollado para optar por el título de : Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 06 de marzo del año 2023

Atentamente,



Stalin Roberto Villalba Armendáriz

1725359606

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Víctor Vinicio Tapia Calvopiña con documento de identificación N° 1708547219, Docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DIAGNÓSTICO DEL ESTADO DE TARJETAS MADRE A TRAVÉS DE INTELIGENCIA ARTIFICIAL Y TERMOGRAFÍA, realizado por Stalin Roberto Villalba Armendáriz con documento de identificación N° 1725359606, obteniendo el como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito,06 de marzo del año 2023

Atentamente,

Ing. Víctor Vinicio Tapia Calvopiña Ms. C.
1708547219

DEDICATORIA

Este trabajo es dedicado primeramente a Dios que me dio salud y fortaleza en este trayecto de mi vida. Dedico mi trabajo a mi abuelita Esperanza Perea por darme la oportunidad de continuar con mi sueño y salir a delante de la mejor forma posible, a mis padres Rolando Villalba y Sara Armendáriz por el apoyo en las ocasiones que más necesitaba y la fortaleza para continuar con mis estudios.

A mis hermanos Gabriel y Mayra por apoyarme en todo este trayecto y la fuerza que me brindaron siempre para poder ser un ejemplo para ellos, a mi cuñado Luis por los consejos y palabras de aliento para continuar mi camino, a mis sobrinos Felipe e Isabela que se sientan muy orgullosos de su tío por cumplir uno de sus sueños, a toda mi familia por brindarme su apoyo incondicional, consejos y palabras para hacer de mí una mejor persona.

AGRADECIMIENTO

Agradezco a Dios por regalarme la vida y permitirme lograr una de mis metas, a mis padres, hermanos, familiares y amigos de la Universidad Politécnica Salesiana que se encontraron siempre apoyándome en todo este trayecto con sus palabras de aliento para no rendirme.

Agradezco a mis mejores amigos Anthony, Eduardo, Jorge, Omar, David, Eduardo, Wendy por siempre estar ahí siendo un apoyo fundamental en todo aspecto, por creer en mi persona, por sus consejos y esas palabras que me ayudaron mucho en las diversas situaciones que se me presentaron en todo este camino.

Agradezco a todo el equipo de Computronik por darme la oportunidad de aplicar mi proyecto en su empresa y del apoyo incondicional por su parte.

Agradezco al Ingeniero Víctor Tapia Tutor en este proyecto, por su tiempo, paciencia y por los consejos en el desarrollo del trabajo de titulación.

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

El realizar un diagnóstico por equipo es aproximadamente de 2 horas en la empresa Computronik tomando en cuenta el número de equipos que ingresan diariamente para su revisión son aproximadamente entre 10 a 15 equipos para su reparación, por este motivo se propuso realizar un dispositivo que ayude en el servicio técnico con el diagnóstico de los equipos de acuerdos a las necesidades requeridas. El dispositivo que ayudará en el servicio técnico está compuesto por una Raspberry, dos cámaras donde se aplicará inteligencia artificial en los dispositivos electrónicos principales de una tarjeta madre y termografía, este dispositivo se sugirió para la reducción de tiempo en el diagnóstico de los equipos, como parte de su desarrollo se utilizó Python para la parte de programación con sus librerías y configuración de las cámaras, se realizaron las pruebas necesarias para verificar su desempeño en el área del trabajo y comprobar la eficiencia del dispositivo.

ABSTRACT

Performing a diagnosis by equipment is approximately 2 hours in the Computronik company, taking into account the number of equipment that enters daily for its review are approximately between 10 to 15 equipment for repair, for this reason it was proposed to make a device that helps in the technical service with the diagnosis of the equipment according to the required needs. The device that will help in the technical service is made up of a Raspberry, two cameras where artificial intelligence will be applied in the main electronic devices of a motherboard and thermography, this device was suggested to reduce time in the diagnosis of the equipment, as part of its development, Python was used for the programming part with its libraries and configuration of the cameras, the necessary tests were carried out to verify its performance in the work area and verify the efficiency of the device.

INTRODUCIÓN

En el presente proyecto se estudia una propuesta experimental y busca nuevas alternativas para realizar el diagnóstico de placas madre de computadora para reducir el tiempo de detectar los daños en las placas madre de los elementos electrónicos que se encuentren comprometidos.

Lo cual utilizaremos la termografía que nos ayudará a determinar la temperatura de las placas madre para identificar las zonas de los daños de esta y proceder con su respectiva reparación.

La inteligencia artificial será utilizada para realizar la detección de imagen de los elementos más principales de las placas madre que son procesador, Bios, Kbc, estos elementos electrónicos son importantes en el circuito de un computador por si diversa funcionalidad.

La aplicación de este proyecto se realizará para brindar información base sobre la temperatura que soportan estos elementos para que el usuario sepa identificar el daño de estos elementos.

Todo este proceso que se aplicará en nuestro proyecto ayuda al desarrollo de la tecnología en diversas ramas, ya que en nuestro país no se emplea este tipo de tecnología en ciertas áreas, el beneficio de este prototipo es su aplicación en el área de reparación enfocado en las placas madre.

1. CAPÍULO 1

1.1 Descripción del problema

En la empresa Computronik el tiempo estimado para un diagnóstico por equipo es de alrededor 2 horas tomando en cuenta la cantidad de equipos que ingresan semanalmente al servicio técnico para su revisión son aproximadamente de 10 a 15 equipos para su reparación, considerando que existen ocasiones donde surge la necesidad, visto desde la empresa como negativo la contratación temporal de personal extra para suplir la demanda de reparación.

Consecuencia de lo antes mencionado, las personas que se encuentra encargadas de este departamento no pueden cumplir con las necesidades inmediatas del cliente, debido al tiempo que se emplea durante el proceso de diagnóstico de los equipos, el cual conlleva (polución, sulfatación de pistas en la placa madre y sobrecalentamiento), y además de considerar las pruebas de funcionamiento.

Esta situación ha llevado que varias empresas desarrollen o adquieran estrategias administrativas o tecnológicas para mejorar este servicio (Sánchez, S.; Granados, A. 2019, p 21-23).

1.2 Importancia y alcances

La importancia de este proyecto es la ayuda que brindará a la persona encargada del servicio técnico, ya que el sistema que va a ser implementado, le dará información base sobre los componentes más importantes de las placas madre (procesador, BIOS, KBC, chip gráfico). Teniendo esta información podrá detectar los daños con mayor facilidad en los objetos mencionados, además ayudará a disminuir el tiempo de diagnóstico del equipo dando así un beneficio a la empresa para su crecimiento.

La empresa Computronik requiere de un sistema que le permita el diagnóstico del estado del funcionamiento de una tarjeta madre, para lo cual este proyecto pretende desarrollar un equipo que contenga cámaras, una fotográfica y una termográfica para la detección de temperatura, utilizando inteligencia artificial. De esta manera dicha empresa quiere ser competitiva a nivel nacional, aplicando nuevas tecnologías en la estación de servicio técnico.

Los alcances del proyecto es poder lo mejorar y aplicar en los diferentes lugares para su uso, no únicamente para reparaciones de placas madre también puede ser aplicado en la industria para la detección de diversos daños en ciertos elementos por la detección de temperatura.

1.3 Delimitación del problema

1.3.1 Temporal

El objeto de la investigación y desarrollo de este proyecto se realizará en la ciudad de Quito, Ecuador en el periodo 2022-2023 y tendrá aproximadamente una duración de 6 meses.

1.3.2 Geográfica

El proyecto será un minicomputador con dos cámaras y podrá ser transportado con total facilidad de una instancia a otra.

1.3.3 Académica

Este proyecto se realizará cumpliendo las normas impuestas por la Universidad Politécnica Salesiana basado en su grado investigativo y modelo de presentación para proyectos de titulación, también se pondrán en práctica los conocimientos adquiridos durante todo el proceso de estudio en materias como: Electrónica digital, Electrónica Analógica, Sistemas Microprocesados I y II, Informática Industrial.

1.4 Objetivos

1.4.1 Objetivo General

Diagnosticar el estado de Tarjetas Madre a través de Inteligencia Artificial y termografía para la identificación de posibles daños electrónicos.

1.4.2 Objetivos Específicos

- Investigar las características de las tarjetas madre, plataformas, librerías y software, para determinar la arquitectura de la red neuronal, a través de artículos y revistas indexadas.
- Desarrollar un algoritmo que permita la identificación de dispositivos fundamentales de la tarjeta madre con exceso de temperatura a través de inteligencia artificial.
- Implementar un sistema del estado de tarjetas madre para su diagnóstico en la estación de servicio técnico de la empresa Computronik utilizando hardware y software especializado.
- Realizar pruebas de funcionamiento sobre el sistema implementado para la verificación de los resultados a través de pruebas experimentales en un entorno controlado.

2. CAPÍTULO 2

Este capítulo se trata sobe la investigación de varios conceptos, para la aplicación del mismos en nuestro proyecto y así tener el conocimiento suficiente sobre los elementos a utilizar en el desarrollo.

4.3 Raspberry PI

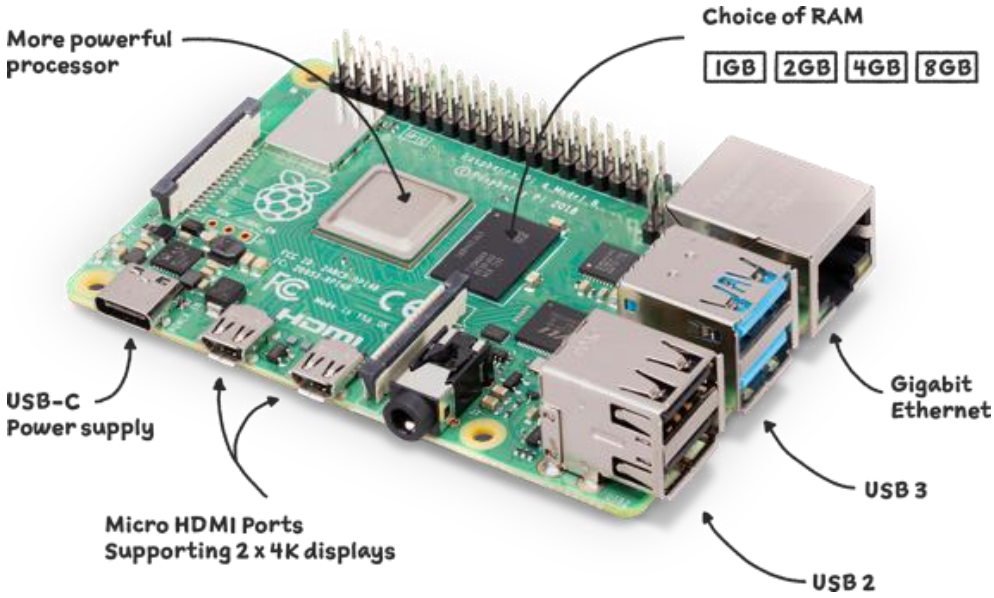
Raspberry pi es una placa base que posee distintos componentes de un ordenador como un procesador ARM 1500 mhz, chip de video y Memoria RAM de hasta 8gb, también tiene distintos puertos de conexión como los siguientes:

- 2 puertos hmi
- Ranura micro sd
- 4 puertos usb
- Puerto ethernet
- Entrada de audio
- Alimentación mediante entrada tipo c
- Entradas que permiten la conexión de módulos especiales para esta tarjeta (pines).
- Conexión wifi y bluethooth.

La primera Raspberry Pi apareció en un grupo de investigación en la UNIVERSIDAD DE CAMBRIGE, para la enseñanza de computadores en los años 90 por los grandes costos de ordenadores (pc), las primeras Raspberrys lanzadas al mercado fue en el año 2012 así siendo utilizada para grandes proyectos por su bajo costo y funcionabilidad con el tiempo a sido mejorada para ser adaptada a los requerimientos de la actualidad.

La Raspberry Pi por ser un minicomputador posee algunas limitaciones como su velocidad de procesamiento para ciertas actividades, no tiene un sistema de apagado porque tiene un consumo bajo de 2,5 amperios y 5 voltios, por el momento el sistema operativo que se utiliza es el Linux, Windows 10 IOT que es especializado para este minicomputador es de 32 bits por el momento ya que se está implementado un sistema para que pueda ser de 64 bits como se lo utiliza en estos tiempos.(Leticia,C, 10 de marzo 2022).

Figura 2.1 Raspberry PI 4



Minicomputador Fuente: (Página Oficial de Rasberry)

2.2 Cámara térmica MLX90640

Esta es una cámara térmica con una matriz IR de 32x24 pixeles, su campo de visión es de 55°, que se comunica por una interfaz i2c, es compatible con las tarjetas Raspberry y Arduino.

Características:

- Funciona con voltajes de 3.3 v a 5v
- Es compatible con Raspberry y Arduino
- Tiene una matriz de sensores térmicos de infrarrojo lejano de 32x24
- Temperatura que detecta es de -40°C a 300°C

(Melexis,2019)

Estos dispositivos para realizar su medición utilizan la ley de radiación de Planck y Boltzmann a través de la radiación infrarroja emitida por el dispositivo a medir.

Al realizar todo este proceso se necesita un lente que enviar una señal de la temperatura del objeto mediante una pantalla. (Melexis,2019)

El sensor infrarrojo recibe esa radiación de la superficie del elemento, pero también refleja del entorno su radiación la cual penetra al objeto así obteniendo un dato de medición en temperatura por el dispositivo. (Melexis,2019)

Figura 2.2 Cámara térmica MLX90640

Cámara para medición de temperatura Fuente: (Melexis,2019)

Esta cámara fue elegida por la disponibilidad y accesibilidad al dispositivo, también por sus características que cumplen con lo requerido para el proyecto, es uno de los módulos que es compatible con nuestra tarjeta, posee un sensor de temperatura y la imagen termográfica que son necesarias para el desarrollo.

2.3 Genius faceCam 1000x

Esta es una webcam con la cual se la puede utilizar con distintos programas esto no quiere decir que no se la pueda utilizar como una cámara de alta calidad ya que sus propiedades son excelentes a continuación mencionaremos sus características:

- Resolución hd 720
- Resolución (DP) 1080 x 720, 640 x 480 pixeles
- Micrófono incorporado
- Formato de archivos avi/wmv

- Zoom digital 3x
- Compatible con sistema operativo: Windows XP, VISTA, 7 o superior, Mac OS 10.4.6 o posterior, Linux 2.6.21 anterior, Puerto USB

Figura 2.3 Cámara Genius 1000x

Webcam a 720p Fuente: (Genius 2019)

Esta cámara la tomamos en consideración para nuestro proyecto, primero por la compatibilidad con nuestra tarjeta Raspberry y por la utilización en realizar detección de imagen en otra plataforma, lo cual el rendimiento es óptimo para realizar este tipo de trabajo aparte de su costo que es económico para las características que ofrece y de la mejor calidad. (Genius 2019).

2.4 Termografía

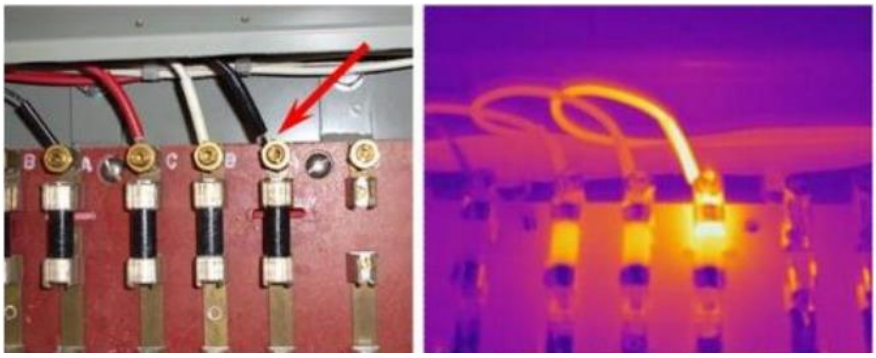
La termografía, nos permite medir temperaturas a distancia y sin necesidad de contacto físico con el objetivo del estudio mediante la captación de la radiación infrarroja del espectro electromagnético.

Todos los objetos tienen información térmica, imperceptible a simple vista pero que se le los puede observar mediante las cámaras termográficas.

La información térmica corresponde a un patrón, un estado puntual en cuanto a su temperatura. Se dice que es puntual porque al objeto no se lo ve como algo aislado, sino al contrario, estará bajo unas condiciones cambiantes, rodeado de otros objetos que le influyen.

La cámara térmica genera imágenes basadas en la temperatura de los objetos, básicamente midiendo la energía infrarroja que emiten y convirtiendo esta información en imágenes cuyos puntos muestran diferentes colores en función de la temperatura superficial de los objetos. (Sa-Nguannarm, P., Charoenpong, T., Chianrabutra, C., & Kiatsoontorn,K.2019).

Figura 2.4 Imágenes de luz Visible y térmica



Comparación con vista infrarroja para medición de temperatura Fuente: (Infrared Traning Center. Course Manual Thermography 2009)

La imagen de la izquierda es una fotografía común que muestra las imágenes de los objetos, es una imagen obtenida con luz visible.

La imagen de la derecha muestra las temperaturas de los objetos, es una imagen termográfica.

Si los niveles de temperatura no son controlados, pueden causar graves daños en los procesos y materiales y por ende en la calidad del producto final. Por ello la importancia de un mapeo de temperatura dentro de la industria, donde los entes reguladores son exigentes y dichos elementos deben ser controlados. (Sa-Nguannarm, P., Charoenpong, T., Chianrabutra, C., & Kiatsoontorn,K.2019).

2.5 Inteligencia artificial

Es la capacidad que posee una máquina para realizar diferentes funciones a partir de los datos y algoritmos aprendidos, en la toma de decisiones como realizaría un ser humano.

Una persona necesita descansar, lo que no ocurre con las maquinas ya que pueden analizar varios volúmenes de información a la vez, también los errores en las máquinas son menores a las que comenten las personas esto quiere decir que son mas precisas por esta razón los sistemas de inteligencia artificial son utilizados para realizar diferentes actividades.

Uno de los beneficios de la inteligencia artificial es que tanto máquinas como robots nos ayudaran a realizar tareas que los humanos consideren de alto riesgo o difíciles, también tenemos que considerar que la inteligencia artificial es un tema muy extenso en lo cual se puede implementar en varias áreas de trabajo desde lo comercial hasta en la industria, el desarrollo de la inteligencia artificial no tiene límite, ya que siempre es aplicable para las necesidades requeridas en distintos lugares de trabajo. (Rouhiainen.L 2018).

2.5.1 Aprendizaje autónomo

El aprendizaje automático o también llamado machine learning es uno de los principales enfoques de la inteligencia artificial, es un aspecto en el cual el sistema informático o las maquinas sin necesidad de estar programados tienen la capacidad de aprender, como ejemplo se puede decir las predicciones o sugerencias en una situación particular.

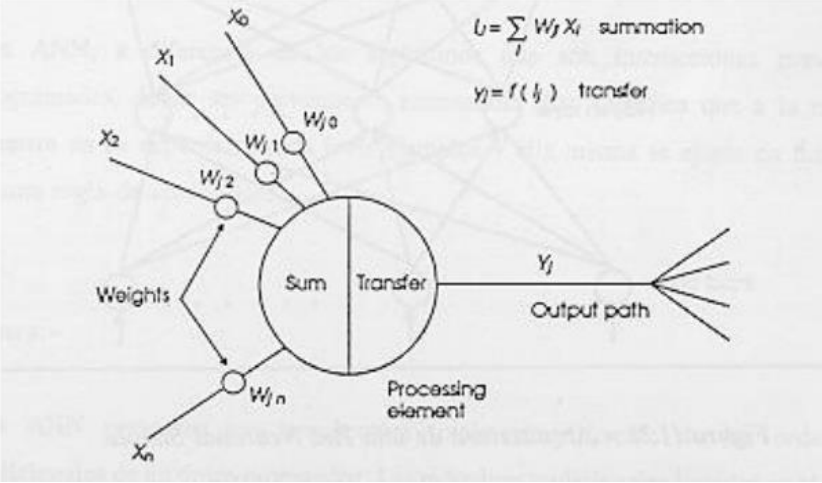
Este tipo de aprendizaje lo que nos ayudará en un futuro en la manera de utilizar cierta herramienta, ira aprendiendo así mejorando la experiencia al usuario de una u otra manera en la respectiva aplicación, así con el tiempo adquiriendo experiencia el aparato al utilizar lo. (Rouhiainen. L,2018).

2.5.2 Red Neuronal Artificial

La red neuronal artificial es una unidad análoga, posee varias entradas y se las combina con una simple suma básica. La suma de las entradas es cambiada por una función de transferencia y el resultado de la salida de esta función va inmediatamente a la salida del elemento del procesador.

La salida se puede conectar a otras neuronas mediante sus entradas realizando conexiones ponderadas correspondientes a las sinapsis neuronales se representa en la siguiente figura. (Basogain. X,2018)

Figura 2.5 Diagrama de una Neurona Artificial



Proceso de una neurona artificial Fuente: (Basogain. X,2018)

2.6 Haar Cascade

Es una herramienta complementaria que ayuda al entrenamiento para el reconocimiento de objetos con la ayuda de lenguaje de programación, librerías de visión artificial (Opencv), esta herramienta nos permite procesar imágenes y así generando un modelo para la identificación o reconocimiento de objeto, clasificarlos y seguirlos.

Se opto por la utilización de esta herramienta más por el método de clasificación que es basada en imágenes positivas (objeto con su entorno) e imágenes negativas (objetos no a

identificar) esto quiere decir que no se tomaran en cuenta las imágenes negativas en el reconocimiento del objeto, sino que las excluirá al ser detectadas.

La aplicación de esta herramienta en mi proyecto fue por su eficiencia al detectar objetos ayudando así con el desarrollo del lenguaje de programación para que sea óptimo en su utilización, también debemos aclarar que el proceso para realizar es más didáctico que otras herramientas como por ejemplo el Tensor Flow. (Jeremías. A, noviembre del 2020).

2.7 Tarjeta Madre

La tarjeta madre, es una tarjeta de circuito impreso que posibilita la integración de todos los componentes de una computadora. Para esto, cuenta con un software básico conocido como BIOS, que le permite cumplir con sus funciones como: el monitoreo, la administración o la gestión de la energía eléctrica, así como la distribución de la misma por todo el computador, la conexión física de los diversos componentes (HMI,ratones, impresoras o un escáner), por supuesto, la temporización y su sincronismo. (Hussein, M.; Mutlag, A. 2019).

2.7.1 Procesador

El procesador de un computador es el dispositivo de hadware que tiene diversas propiedades, también se le conoce como el cerebro del sistema del CPU.

Un procesador está constituido por varios núcleos, cache, controladores de memoria, tarjeta gráfica y otros elementos.

Este elemento soporta hasta 0 °C a 90°C cuando un componente como este sobrepasa su temperatura máxima puede llegar a dañarse por esa misma razón este componente tiene varios elementos de protección como disipador, pasta térmica y el sistema de auto apagado que posee la placa madre. (concepto definición, 2022).

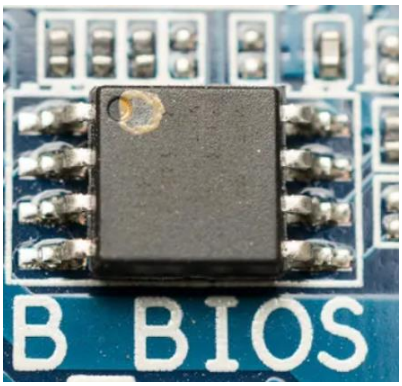
Figura 2.6 Procesador

Cerebro de la computadora Fuente: (Ranchal. J, 24 de marzo del 2018)

2.7.2 BIOS

Es el elemento electrónico encargado de las entradas y salidas de software como también del arranque del propio computador, también es el encargado del flujo de datos del sistema operativo, este tipo de dispositivo su temperatura se encuentra en el rango de 0°C a 70°C. (Página oficial HP, 2022)

Figura 2.7.1 BIOS



Sistema de encendido y de comprobación Fuente: (Naranjo. M, 23 Abril del 2022)

2.7.3 KBC

Las siglas KBC significan keyboard connector, este elemento se encarga de controlar la temperatura, energía que es repartida por la tarjeta madre es uno de los dispositivos más importantes del computador, posee 128 pines su temperatura es de 0°C a 70°C. (datasheetcafe, 19 de abril 2022).

Figura 2.7.2 KBC

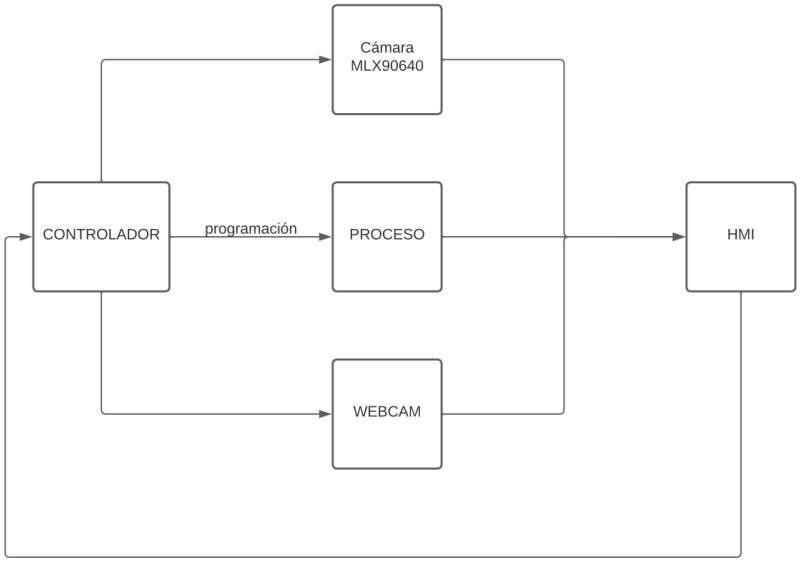
Controlador de voltajes Fuente: datasheetcafe, 19 de Abril 2022)

3. CAPÍTULO 3

En este capítulo indicamos todo el proceso que tuvimos que realizar para lograr los objetivos con nuestro prototipo con su respectiva programación y accesorios.

4.3 Diagrama de bloques

Figura 3.1 Diagrama de bloques de funcionamiento



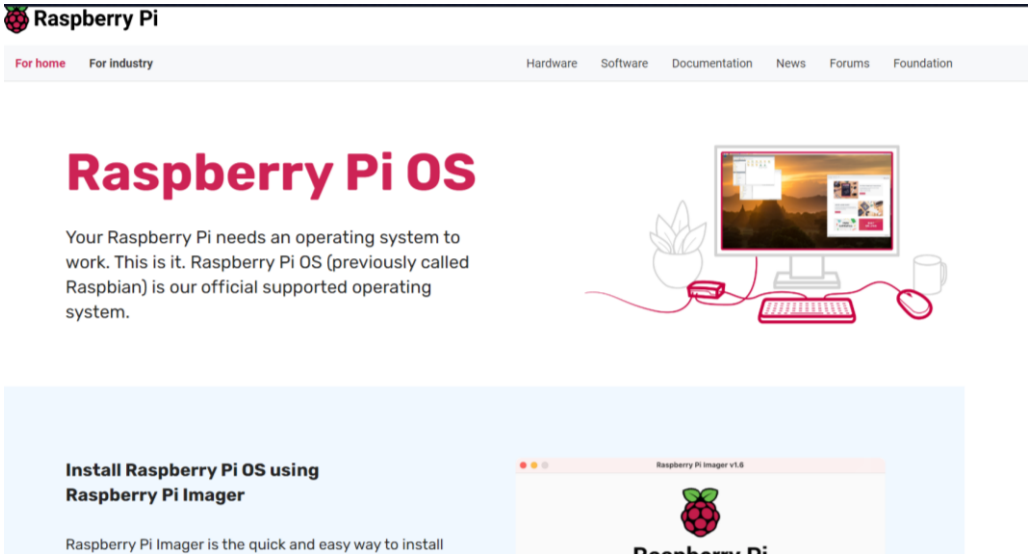
Explicación del funcionamiento del proyecto Fuente: (Villalba. S,2022)

Como se puede observar en la figura se indica el funcionamiento del prototipo de una manera más didáctica en el siguiente punto se explica cómo se fue realizando el prototipo.

3.2 Instalación de sistema operativo y librerías

Para realizar la programación del prototipo primero debemos configurar nuestra tarjeta Raspberry pi 4B lo cual necesitamos instalar su sistema operativo que se encuentra en su sitio web como se observa la en la figura.

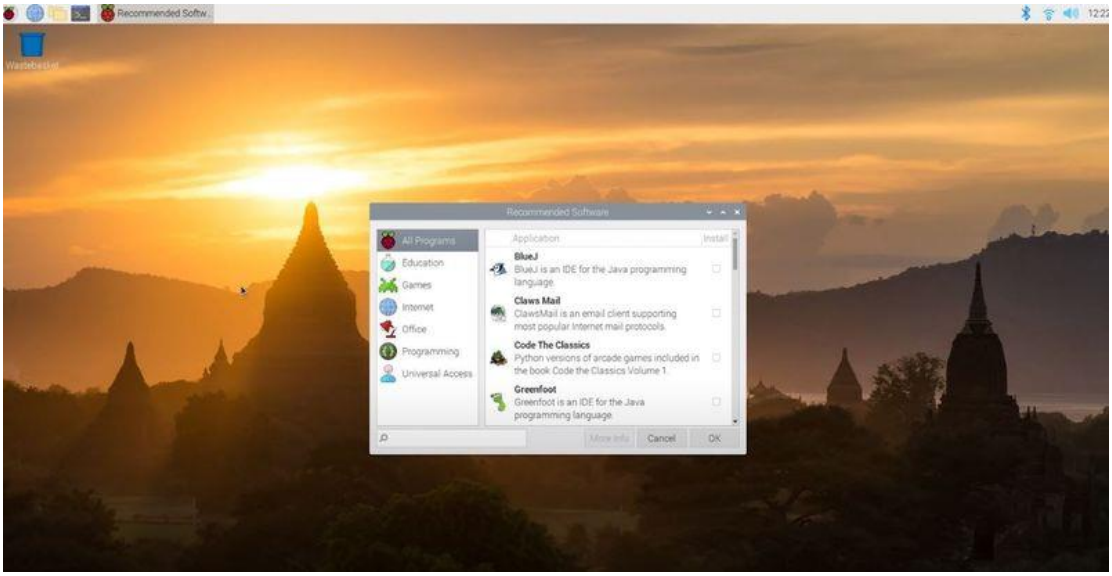
Figura 3.2.1 Interfaz de la página oficial de Raspberry



Pagina donde se obtiene el ios para la instalación del sistema operatico Fuente: (Villalba. S,2022)

Aquí debemos tener en cuenta que existen tres tipos de sistemas operativos, donde ya tienen programas preinstalados en este caso se prefirió ir por el sistema operativo que tenía los programas básicos.

Figura 3.2.2 Escritorio de la tarjeta Raspberry



Se observa en funcionamiento el sistema operatico Fuente: (Villalba. S, 2022)

Como se puede apreciar en la figura la interfaz de la Rasberry con el sistema operativo por defecto que es el Linux, como dato importante hay que añadir que, para realizar la programación adecuada de este proyecto se debe instalar Opencv el cual ayudará para poder obtener video de las distintas cámaras.

```
Sudo apt-get install python3-opencv
```

esta línea de código se debe ingresar en el Cmd de la Raspberry ya que es la única manera de instalar distintas librerías y programas para Python, al utilizar este comando nos instalar el Opencv pero un dato que se debe tener en cuenta es que no se instala con las últimas versiones lo cual hay que instalar manualmente las distintas librerías del Opencv para su perfecto funcionamiento.

```
Import cv2 as cv
print(cv.__version__)
```

primero en el terminal al escribir estos comandos nos ayudaran para identificar que versión de Opencv contiene nuestra tarjeta Raspberry instalada.

```
Sudo apt-get install cmake
sudo apt-get install gcc g++
sudo apt-get install python3-dev python3-numpy
```

al instalar estos programas los cual nos ayudaran para configurar la instalación, compilación y para crear enlaces en Python.

```
Sudo apt-get install libgtk2.0-dev
sudo apt-get install libgtk-3-dev
```

A continuación, necesitamos compatibilidad con GTK para funciones GUI, compatibilidad con cámara (v4l), compatibilidad con medios (ffmpeg, gstreamer).

Sudo apt-get install libpng-dev
sudo apt-get install libjpeg-dev
sudo apt-get install libopenexr-dev
sudo apt-get install libtiff-dev
sudo apt-get install libwebp-dev

OpenCV viene con archivos de soporte para formatos de imagen como PNG, JPEG, JPEG2000, TIFF, WebP, etc. Pero puede ser un poco antiguo. Si desea obtener las bibliotecas más recientes, puede instalar archivos de desarrollo para bibliotecas del sistema de estos formatos.

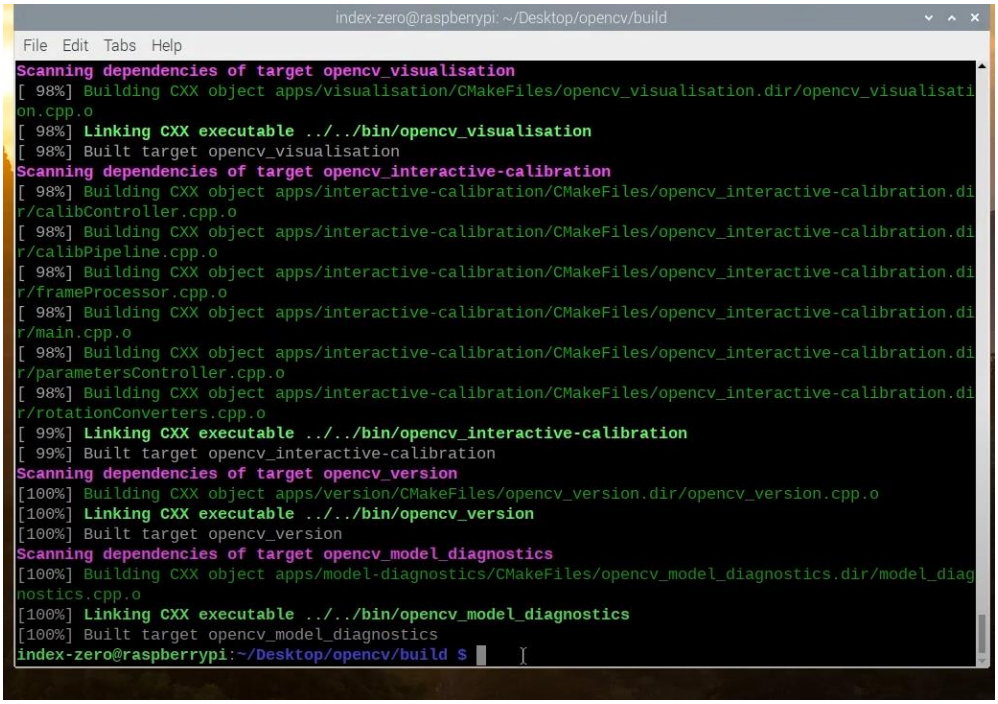
Sudo apt-get install git
git clonar https://github.com/opencv/opencv.git

Creará una carpeta “opencv” en el directorio actual. La clonación puede tardar algún tiempo dependiendo de su conexión a Internet.

Make
sudo make install

al aplicar estos comandos se instalará totalmente el Opencv lo cual llevará un tiempo aproximado de 3 horas, por la velocidad de la tarjeta Raspberry.

Figura 3.2.3 Terminal de la Raspberry instalación Opencv

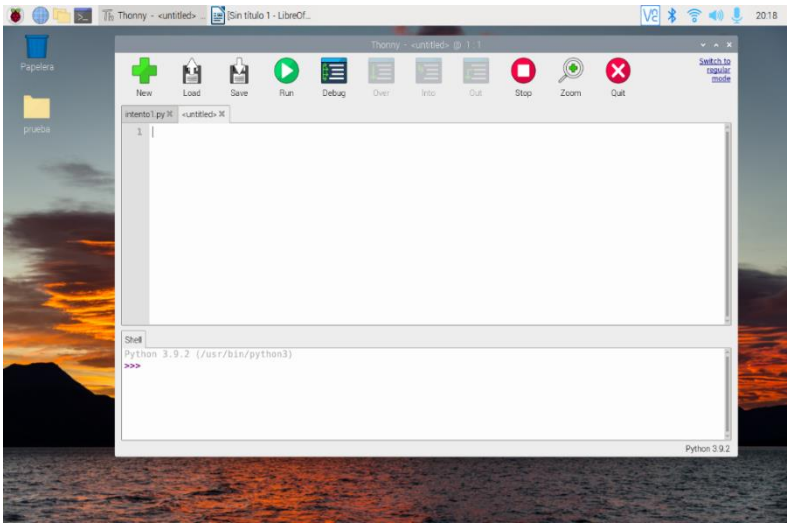


Aquí se observa la instalación del porcentaje del Opencv Fuente: (Villalba. S, 2022)

Como se observa en la figura 3.2.3 es la instalación de Opencv al terminar este proceso se puede utilizar para el siguiente paso en el avance de este proyecto.

3.3 Etapa 1: Desarrollo de captura para el entrenamiento de la neurona

Figura 3.3.1 Programa Thonny Python



Interfaz de la aplicación para la programación Fuente: (Villalba. S, 2022)

Como se observa en la figura procedimos a utilizar una de las varias interfaces que posee Python en este caso utilizamos el Thonny Python IDE para continuar con esta programación.

```
Import cv2

cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH,640)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT,480)
```

La primera línea de nuestro programa es realizar el llamado de la librería Opencv, después dimensionamos la pantalla donde se va a observar lo que capte nuestra webcam es este caso.

X1, y1 = 250, 170
x2, y2 = 350, 270

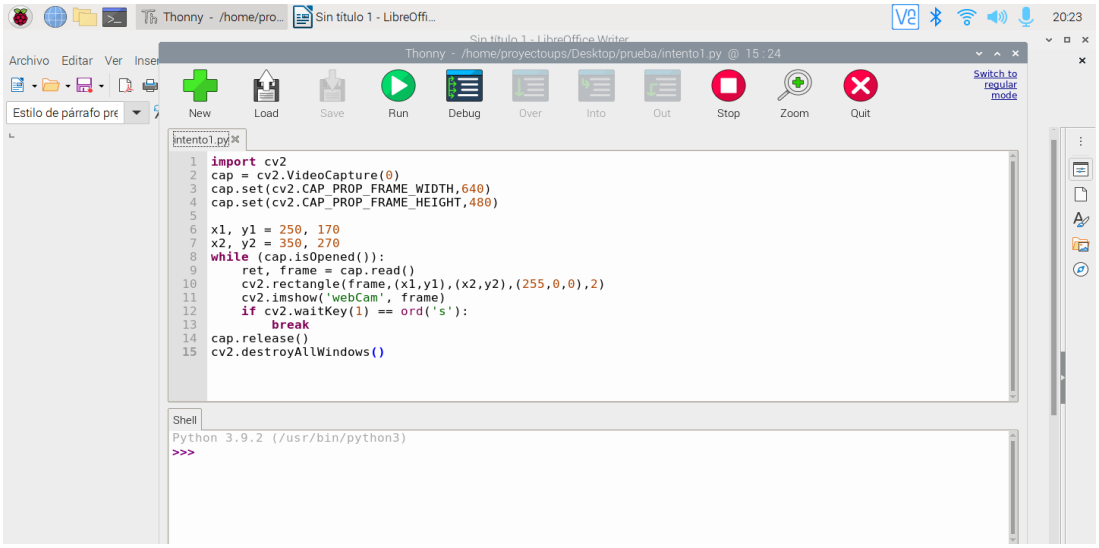
Aquí realizamos la dimensión del cuadrado el cual enfocaremos el objeto para su siguiente etapa de la programación.

```
While (cap.isOpened()):
ret, frame = cap.read()
cv2.rectangle(frame,(x1,y1),(x2,y2),(255,0,0),2)
cv2.imshow('webCam', frame)
if cv2.waitKey(1) == ord('s'):
break
cap.release()
cv2.destroyAllWindows()
```

en esta parte del código aquí abrimos un bucle el cual hacemos que lea la variable cap con el Opencv, también se inserta un cuadrado para que se observe al proyectarse la

llamada de la cámara y para salir del bucle debemos presionar la letra s para cerrar la ventana emergente cuando se corra el programa como se observa en la figura 3.3.2.

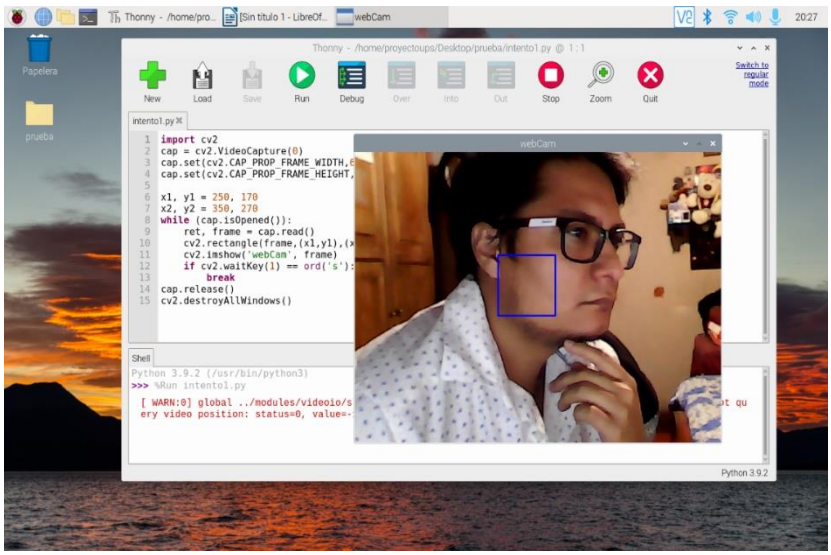
Figura 3.3.2 Programación base



Código realizado en la interfaz Fuente: (Villalba. S, 2022)

a continuación, mandaremos a ejecutar la programación realizada hasta el momento para ver su desempeño como se puede observar en la siguiente figura 3.3.3.

Figura 3.3.3 Funcionamiento de la programación base



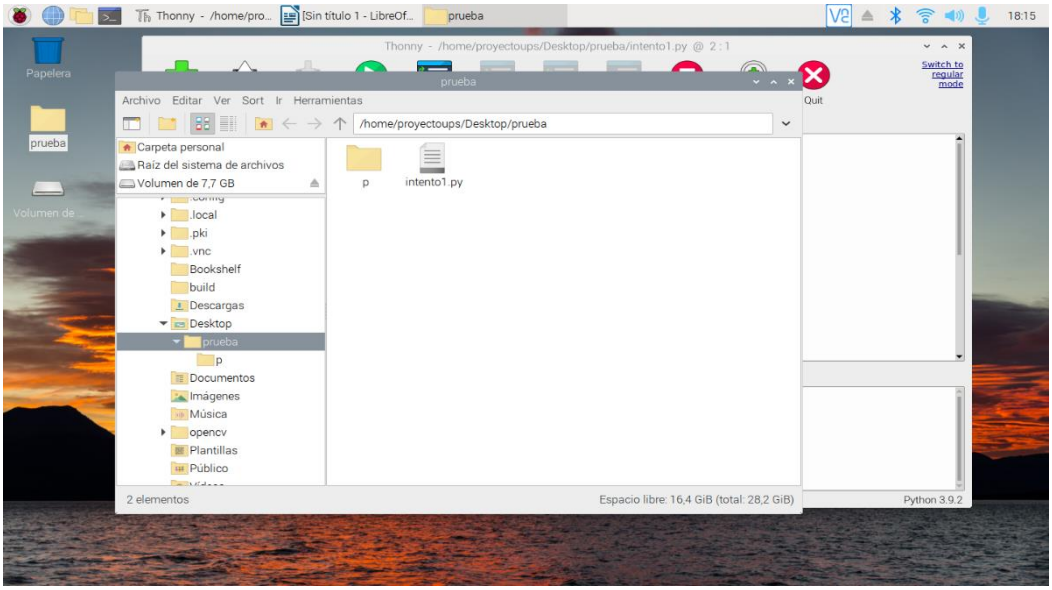
Funcionamiento del método de captura Fuente: (Villalba. S, 2022)

el siguiente paso que realizaremos son las capturas con su debido recorte de imagen lo cual ayudará para su próximo paso, el código que aplicamos son los siguientes:

```
import numpy
import imutils
import os
img_counter = 0
Datos=’p’
if not os.path.exists(Datos):
print(‘Carpeta creada:’ , Datos)
os.mkdir(Datos)
```

Se realizó el llamado de tres librerías las que ayudará en las diversas aplicaciones, como se puede observar creamos dos variables una que será el contador para el número de capturas que realizaremos, la otra variable es para crear una carpeta donde serán almacenadas las capturas realizadas, el resto del código como se puede observar decimos si no se encuentra la carpeta “datos” será creada en el lugar donde esta guardado nuestro código base como se observa en la siguiente figura 3.3.4.

Figura 3.3.4 Creación de carpeta “p” mediante programación



Lugar donde se depositarán todas las capturas positivas Fuente: (Villalba. S, 2022)

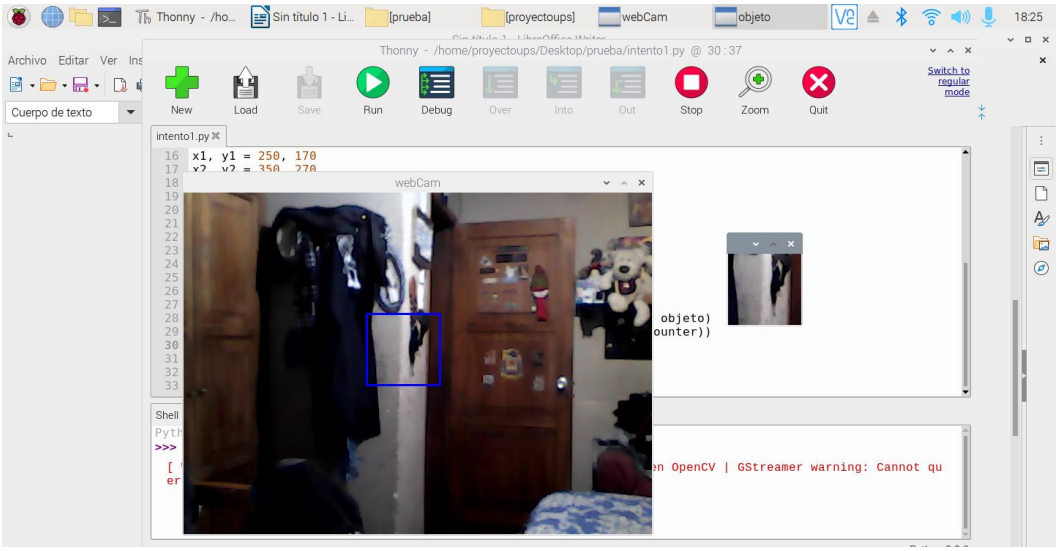
En las siguientes líneas de código indicamos que se realice únicamente la captura de la zona del rectángulo con una medida de la imagen que es de 100 como se indica a continuación en la figura.

imAux= frame.copy()

objeto=imAux[y1:y2,x1:x2]

objeto=imutils.resize(objeto, width=100)

Figura 3.3.5 Recorte de la imagen captada



La imagen al ser capturada será recortada solamente en la sección indicada Fuente: (Villalba. S, 2022)

Indicamos con el comando que al presionar la letra “s” se realizará la captura de la imagen recortada directamente a la carpeta “p”, aquí ponemos que el contador aumente en 1 así dándonos el número de capturas que realicemos como se observa en la siguiente figura 3.3.5.

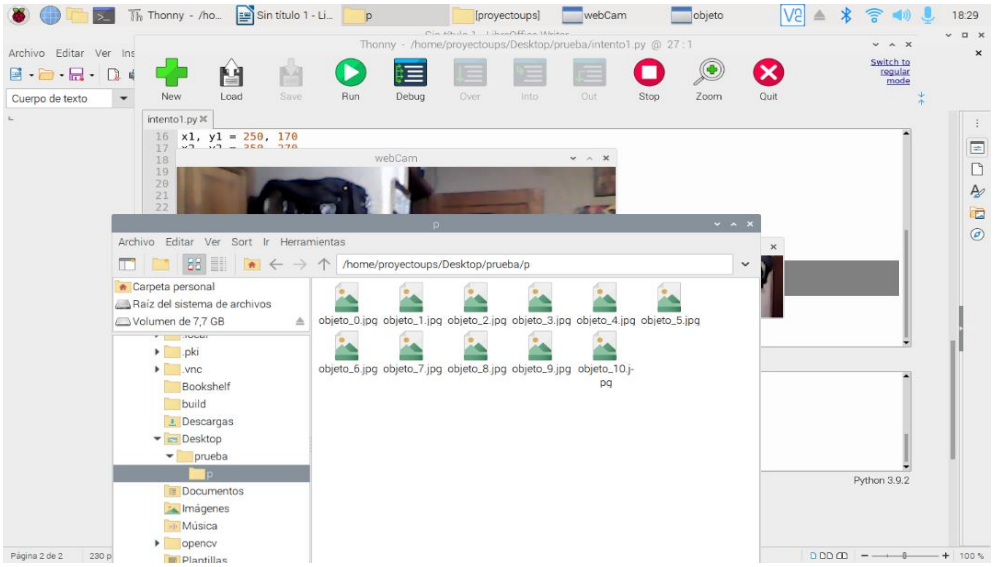
if cv2.waitKey(1) == ord(‘s’):

cv2.imwrite(Datos+’/objeto_{}.jpg’.format(img_counter), objeto)

print(“Imagen guardada: ‘+’/objeto_().jpg”.format(img_counter))

img_counter = img_counter +1

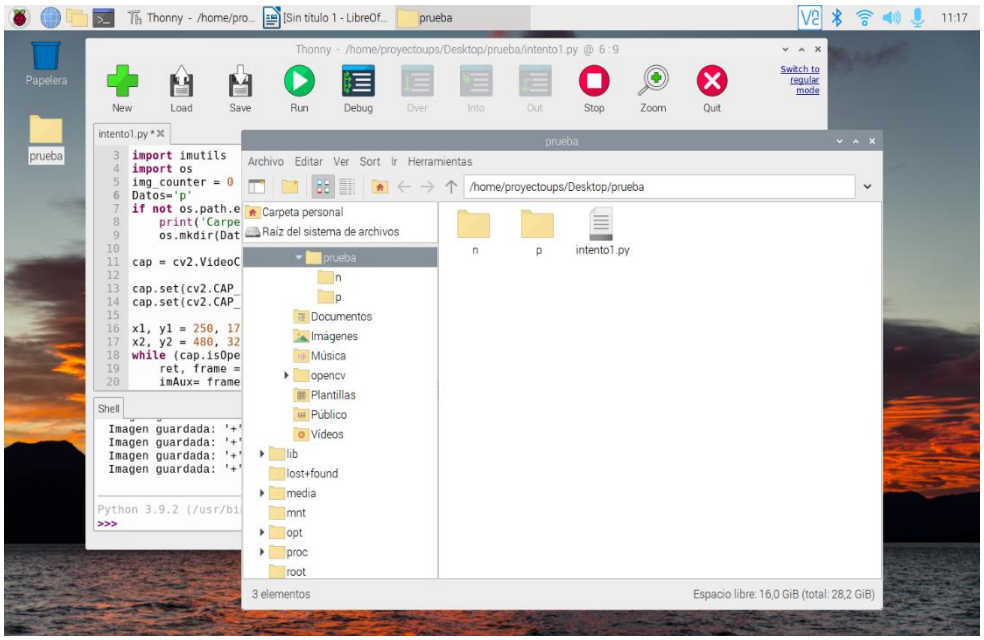
Figura 3.3.6 Imágenes capturadas y almacenadas en la carpeta “p”



Carpeta donde se almacenan las imágenes positivas para el entrenamiento de la neurona Fuente: (Villalba. S, 2022)

Al tener almacenado ya las imágenes debemos hacer unos cambios en nuestros códigos ya que necesitamos capturas tanto como positivas del entorno como negativas del mismo así que debemos crear otra carpeta de datos para realizar las capturas negativas necesarias, esto nos servirá para poder entrenar nuestra neurona y aplicarla en un nuevo código.

Figura 3.3.7 Creación de la carpeta “n” para imágenes negativas



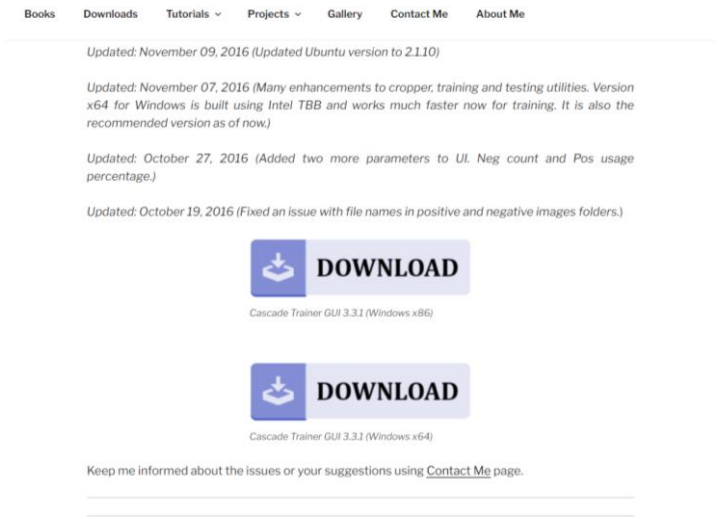
Carpeta donde se guardarán las imágenes negativas Fuente: (Villalba. S, 2022)

Creamos una carpeta n la cual contendrá la captura de la zona negativa del componente que queremos que identifique, lo cual ayudará con más facilidad reconocer este componente cambiando únicamente el nombre de la carpeta en nuestro código inicial.

3.4 Instalación del programa Cascade trainer gui

Ingresamos en la página oficial de cascade donde descargaremos el instalador de la aplicación como se observa en la figura 3.4.1.

Figura 3.4.1 Interfaz de descarga para el entrenador en cascada



Descarga de la aplicación en la página oficial Fuente: (Villalba. S, 2022)

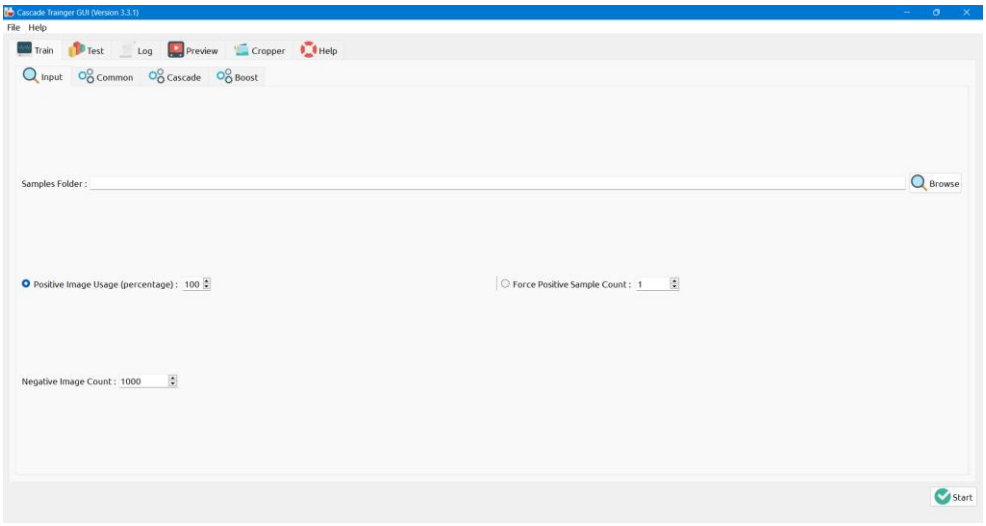
primero realizamos la descarga del instalador del cascade trainer gui para la instalación en este equipo hay que aclarar que este programa solo funciona con el sistema operativo Windows, al tener ya el instalador se le ejecutará y se instalará como un programa normal al terminar la instalación en el escritorio del computador se observará el icono del acceso directo como se muestra en la figura 3.4.2.

Figura 3.4.2 Acceso directo del entrenador cascade

Imagen del programa ya instalado en el computador Fuente: (Villalba. S, 2022)

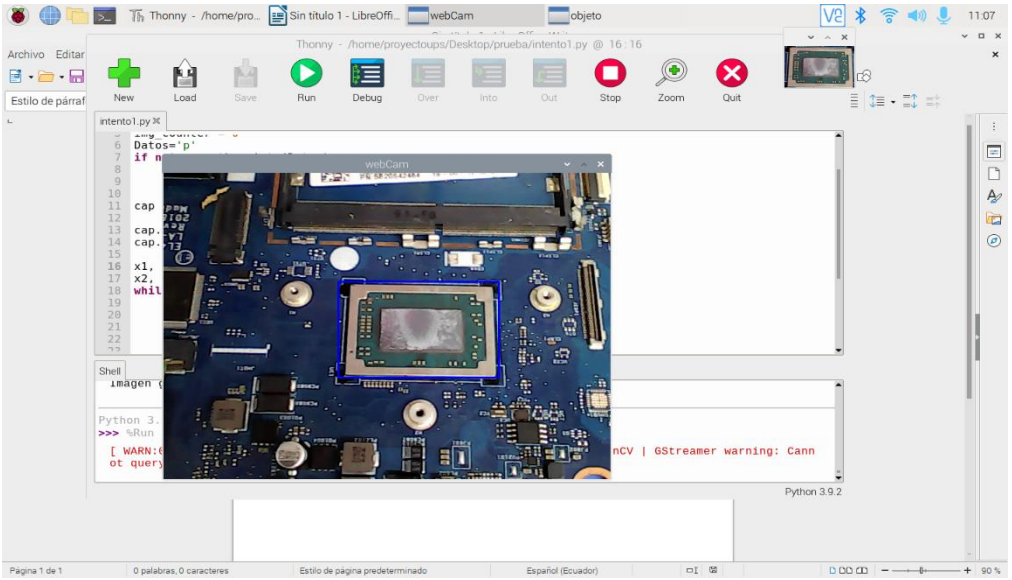
Como siguiente paso realizamos los cambios para la toma de las muestras de uno de los elementos electrónicos que se aplicaran a nuestro proyecto para el aprendizaje de la neurona, Como se muestra en la figura 3.4.3.

Figura 3.4.3 Interfaz cascade trainer-gui



Pantalla principal del trainer-gui Fuente: (Villalba. S, 2022)

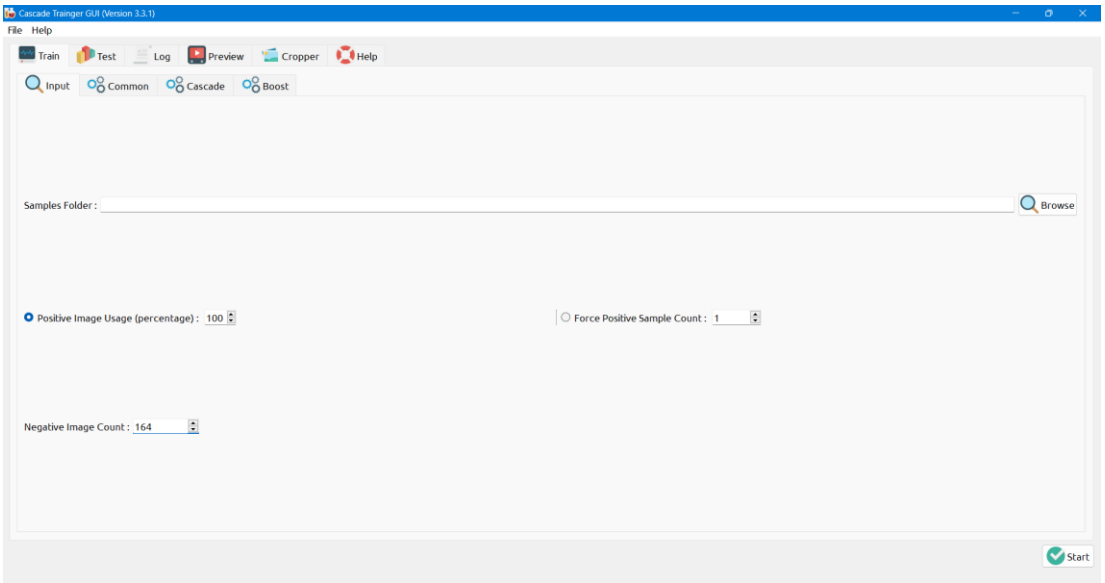
Figura 3.4.4 Captura del procesador



Toma de las imágenes positivas para el entrenamiento de la neurona Fuente: (Villalba. S, 2022)

Utilizando el programa cascade trainer gui para el entrenamiento de la neurona como se observa en la imagen aquí debemos poner la cantidad de tomas positivas como negativas poniendo su cantidad en el entrenamiento al 100% nos indica que la cantidad establecida de fotos positivas son de 40 tomas y para las tomas negativas se debe poner la cantidad que se realizaron en nuestro caso 164 como se muestra en la figura 3.4.4.

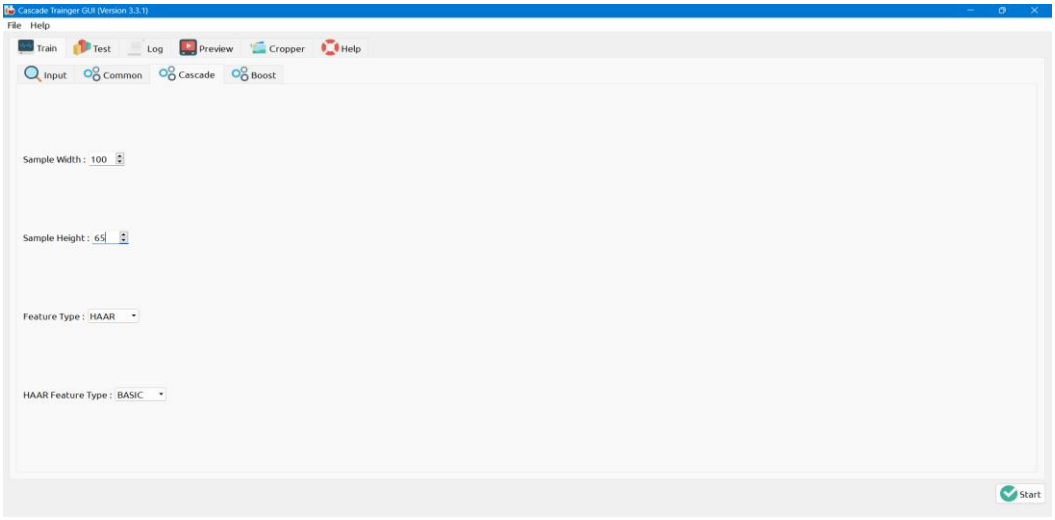
Figura 3.4.5 Cantidad de imágenes positivas y negativas



Configuración del interfaz para el entrenamiento de la neurona Fuente: (Villalba. S, 2022)

En esta sección se configura poner el alto y el ancho de nuestras imágenes que obtuvimos anteriormente para comenzar con el aprendizaje de nuestra neurona como se puede observar en la figura 3.4.6.

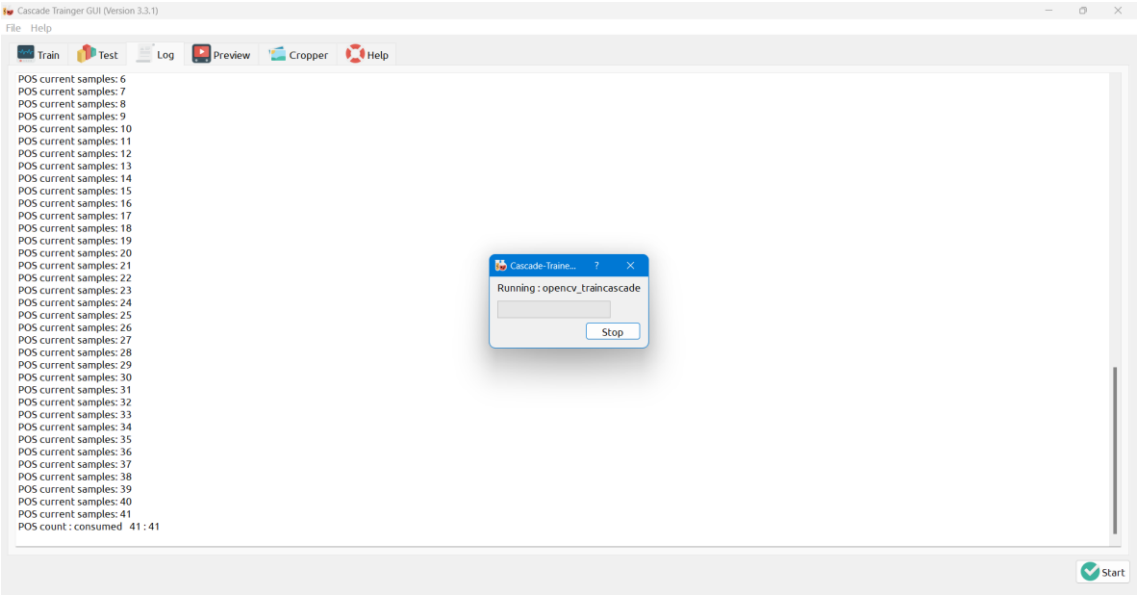
Figura 3.4.6 Dimensiones de las imágenes



Se agrega las dimensiones de las imágenes tomadas para su entrenamiento Fuente: (Villalba. S, 2022)

Aquí se observará el aprendizaje de nuestra neurona mediante el programa cascade trainer gui esto llevará un tiempo en realizarse así el programa emitiendo un archivo Xlm donde se encontrará el entrenamiento de nuestra neurona, así procediendo con el siguiente paso que se mostrará a continuación.

Figura 3.4.7 Interfaz de entrenamiento de la neurona



Proceso de entrenamiento de la neurona Fuente: (Villalba. S, 2022)

3.5 Etapa 2: Desarrollo de la programación de las diferentes neuronas

Como se observa el nuevo código que realizamos es idéntico al que realizamos con anterioridad lo único que utilizamos en este momento es la librería del Opencv también realizamos el llamado de nuestro archivo Xlm que contiene el entrenamiento de nuestra neurona, en el código de este caso como el primer elemento que se realizó en el entrenamiento se lo nombró procesadorClassif que es una variable nueva para nuestra nueva codificación.

```
Import cv2
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH,640)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT,480)
procesadorClassif = cv2.CascadeClassifier('cascade.xml')
x1, y1 = 250, 170
x2, y2 = 480, 320
```

En esta parte del código primero realizó la activación de la webcam en nuestro caso, existen nuevas líneas de código, lo que significan es el cambio de la imagen en grises

para el reconocimiento de objetos, como segundo plano realizamos el llamado de la función detección de imagen en cascada del propio Python donde se codifica el factor de escala de la neurona como el número de vecindades de nuestra neurona.

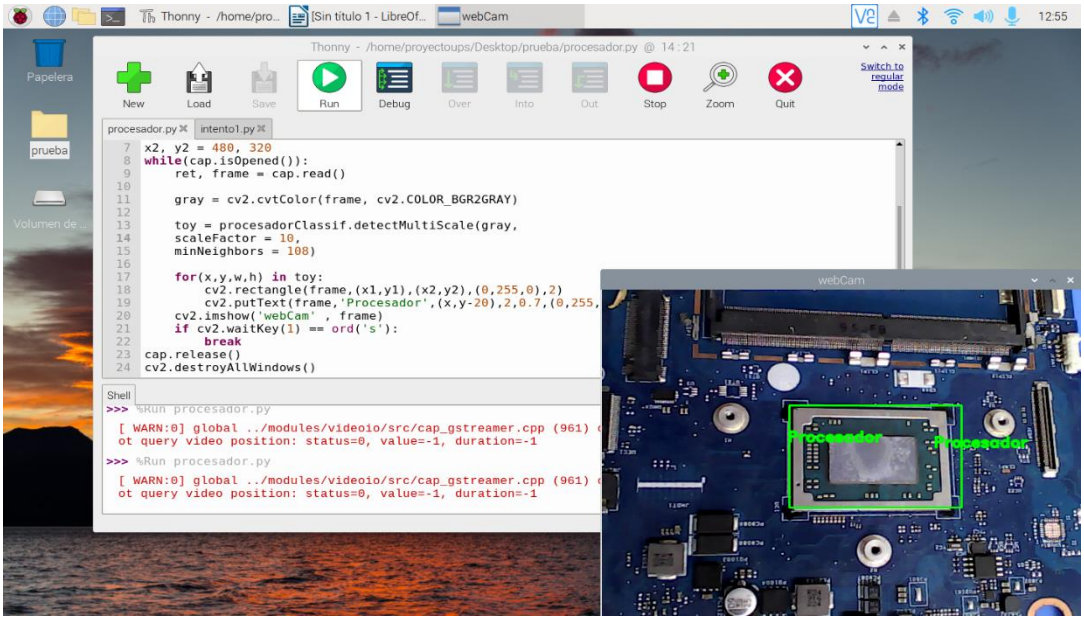
```
While(cap.isOpened()):
ret, frame = cap.read()
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
toy = procesadorClassif.detectMultiScale(gray,
scaleFactor = 10,
minNeighbors = 108)
```

En esta parte del código utilizamos un For donde decimos los parámetros x,y,w,h de nuestra variable “ toy ”, también insertamos un rectángulo para cuando se detecte el objeto se observará en pantalla, se agregó un texto el cual estará nombrando el tipo de objeto y como ultimo presionando la tecla s se cerrara nuestras ventanas al ya no utilizar el programa.

```
For(x,y,w,h) in toy:
cv2.rectangle(frame,(x1,y1),(x2,y2),(0,255,0),2)
cv2.putText(frame,’Procesador’,(x,y20),2,0.7,(0,255,0),2,cv2.LINE_AA)
cv2.imshow(‘webCam’ , frame)
if cv2.waitKey(1) == ord(‘s’):
break
cap.release()
cv2.destroyAllWindows()
```

Como parte final se puede observar el programa en funcionamiento hay que aclarar que el número de escala y de vecindades hay que establecerlas para tener el mínimo error en la detección como se puede observar en la figura 3.5.1, se detecta 2 veces es por el error de la configuración que establecí lo cual en la parte de experimentación se realizará los arreglos necesarios para quitar este error.

Figura 3.5.1 Funcionamiento de la neurona del procesador



Funcionamiento del primer entrenamiento de la neurona Fuente: (Villalba. S, 2022)

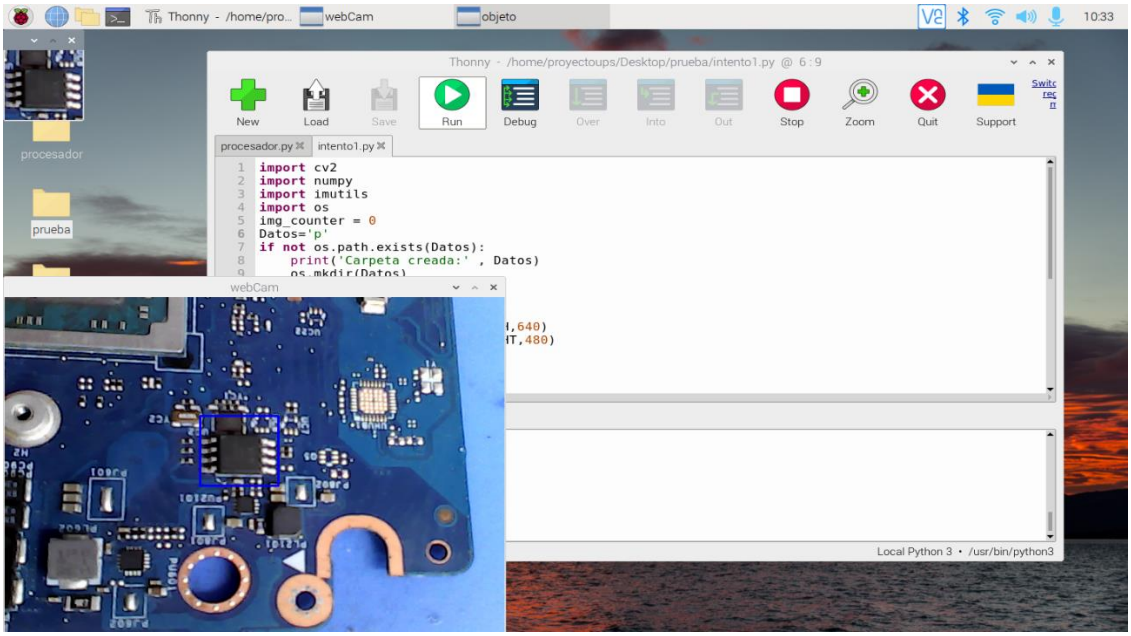
el aprendizaje de la segunda neurona referente al componente electrónico KBC como se puede observar en la figura 3.5.2 los cambios fueron mínimos se utilizó la misma programación realizada con anterioridad los únicos cambios que se realizó fueron crear nuevas variables para el llamado del archivo Xlm para ver su funcionamiento como se observa.

Figura 3.5.2 Funcionamiento de la neurona del KBC

Detección del KBC con la segunda neurona entrenada Fuente: (Villalba. S, 2022)

Como paso siguiente se realizó la toma de muestras del componente electrónico para el entrenamiento de la neurona y mirar el funcionamiento del mismo.

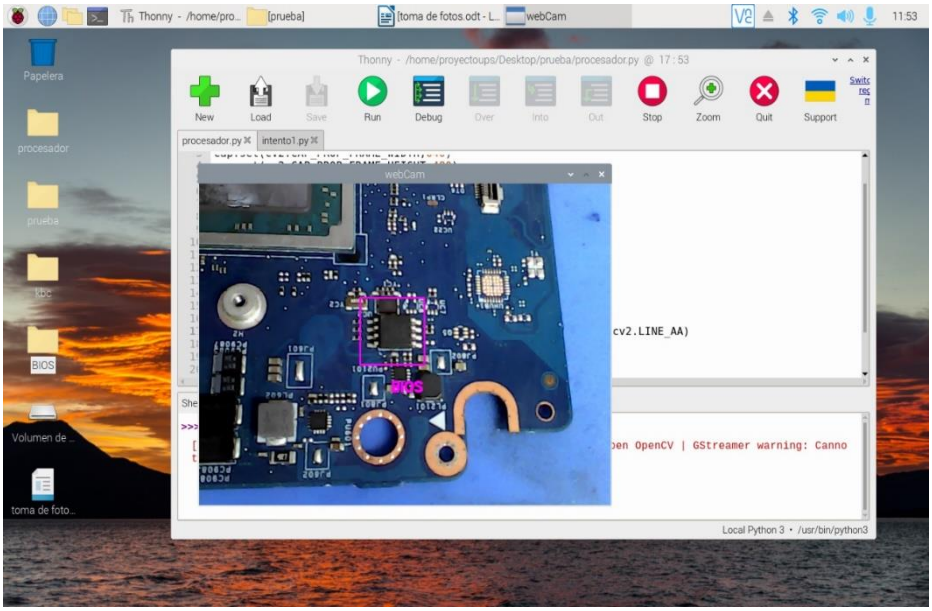
Figura 3.5.3 Capturas de la BIOS



Capturas para el entrenamiento de la tercera neurona Fuente: (Villalba. S, 2022)

Como se observa en la figura 3.5.4 el entrenamiento de la neurona ha sido exitosa lo cual no debemos realizar ningún cambio en la cobertura de área por la cámara.

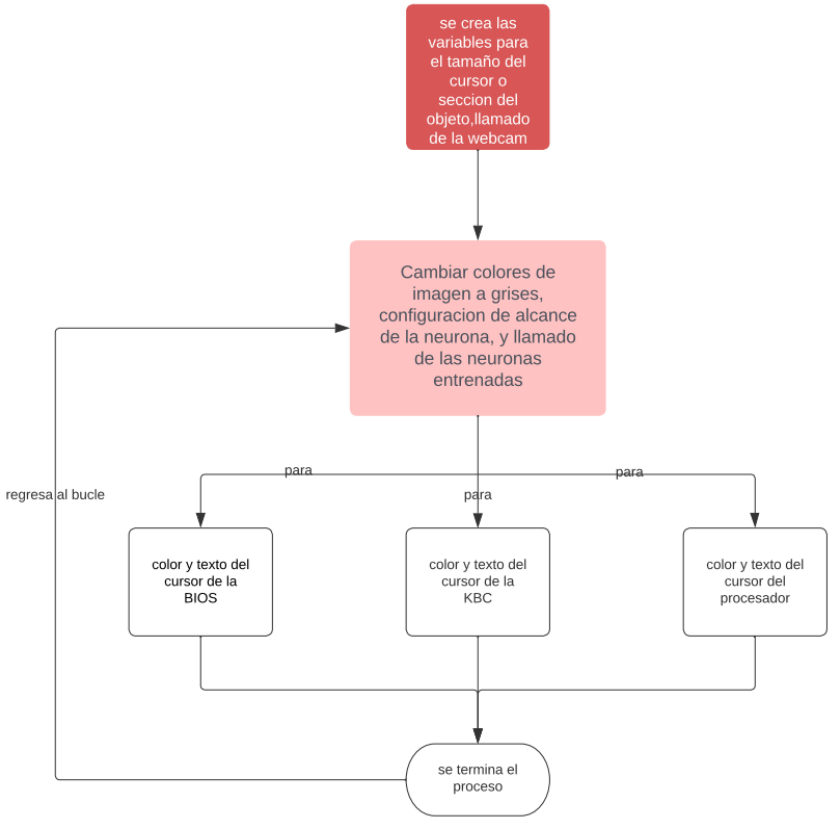
Figura 3.5.4 Funcionamiento de la neurona de la BIOS



Detección de la BIOS con la segunda neurona entrenada Fuente: (Villalba. S, 2022)

3.6 Diagrama de flujo de la programación de todas las neuronas

Figura 3.6.1 Diagrama de flujo de la programación de las neuronas



Explicación de la programación de las neuronas mediante un diagrama de flujo Fuente: (Villalba. S, 2022)

Como se observa en el diagrama se encuentra el procedimiento de la segunda etapa del programa donde se indica la secuencia que sigue este programa a continuación, se explicará cómo se realizó la programación respectiva.

3.6.1 Unión de todas las neuronas entrenadas

Como paso siguiente realizamos la unión de nuestras neuronas en el programa para observar su funcionamiento en conjunto con su respectiva explicación del código

```
biosClassif = cv2.CascadeClassifier('BIOS.xml')
kbcClassif = cv2.CascadeClassifier('KBC.xml')
procesadorClassif = cv2.CascadeClassifier('cascade.xml')
```

en esta parte de la programación lo que realizamos es el llamado de los archivos donde está el entrenamiento de nuestras neuronas de los distintos objetos (KBC, BIOS, Procesador)

x1, y1 = 250, 170
x2, y2 = 480, 320

x3, y3 = 250, 170
x4, y4 = 350, 270

x5, y5 = 250, 170
x6, y6 = 350, 270

aquí lo que establecemos es el tamaño de nuestro marco o cursor de referencia para cada uno de estos objetos a detectar.

```
While(cap.isOpened()):
   ret, frame = cap.read()

   gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

en esta parte del código lo que realizamos es la transformación de la imagen en grises y la lectura de estos archivos mencionados anteriormente.

```
   Procesador = procesadorClassif.detectMultiScale(gray,
   scaleFactor = 25,
   minNeighbors = 225)

   kbc = kbcClassif.detectMultiScale(gray,
   scaleFactor = 20,
   minNeighbors = 200)

   bios = biosClassif.detectMultiScale(gray,
   scaleFactor = 3,
```

```
    minNeighbors = 280)
```

en esta parte de la codificación aplicamos la calibración de cada una de estas neuronas de acuerdo con su factor de escala.

```
    For(x,y,w,h) in procesador:
        cv2.putText(frame,'Procesador',(x,y-20),2,0.7,(0,255,0),2,cv2.LINE_AA)
        cv2.rectangle(frame,(x1,y1),(x2,y2),(0,255,0),2)

    for(x,y,w,h) in kbc:
        cv2.putText(frame,'KBC',(x,y-20),2,0.7,(0,0,255),2,cv2.LINE_AA)
        cv2.rectangle(frame,(x3,y3),(x4,y4),(0,0,255),2)

for(x,y,w,h) in bios:
        cv2.putText(frame,'BIOS',(x,y-20),2,0.7,(255,0,255),2,cv2.LINE_AA)
        cv2.rectangle(frame,(x5,y5),(x6,y6),(255,0,255),2)
```

aquí lo que aplicamos es un For en cada una de estas neuronas para establecer el color del texto y el contorno de la imagen captada así sea cuadrada o rectangular dependiendo las dimensiones que se establecieron.

3.7 Etapa 3: Configuración de la cámara térmica MLX90640 y programación

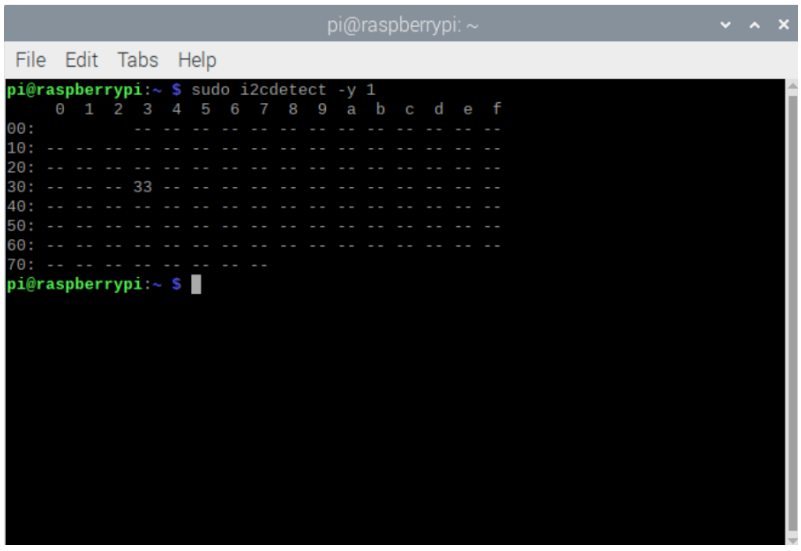
A continuación, iniciaremos con la instalación del módulo térmico MLX90640 el cual ayudará a identificar las zonas de más calor para su instalación necesitamos las siguientes librerías que deben ser aplicadas desde el terminal de la Raspberry.

```
Sudo pip3 install matplotlib scipy numpy
sudo apt-get install -y python-smbus
sudo apt-get install -y i2c-tools
```

Al conectar el módulo en los pines respectivos con el siguiente comando se identifica si está conectado respectivamente con la Raspberry como se observa en la siguiente figura 3.7.1 con su respectivo comando.

```
Sudo i2cdetect -y 1
```

Figura 3.7.1 Verificación del puerto i2c conectado a la cámara MLX90640



Función del puerto i2c de la tarjeta RASPBERRY Fuente: (Villalba. S, 2022)

Después de realizar el paso anterior instalaremos unas librerías que son muy importantes la cual nos ayudara para la imagen de la cámara MLX90640.

Sudo pip3 install RPG.GPIO adafruit-blinka

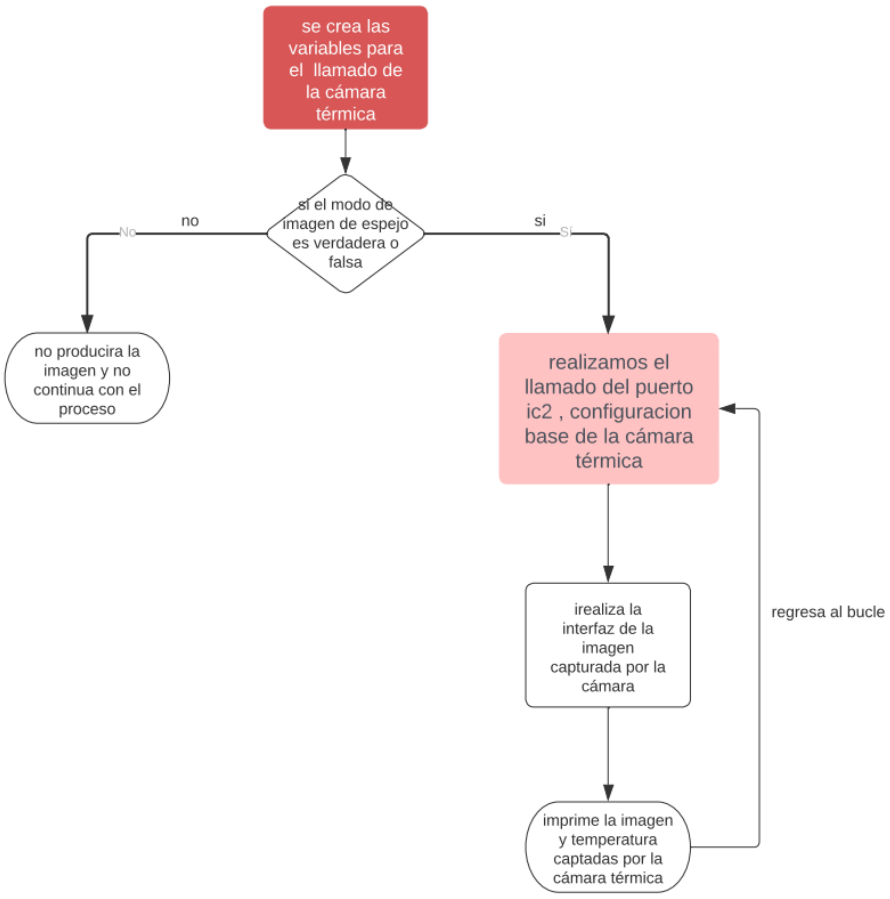
Sudo pip3 install adafruit-circuitpython-mlx90640

Sudo apt-get install idle3

Al instalar las librerías necesarias podemos utilizar la cámara MLX90640 hay que añadir que las dificultades para utilizar este módulo son varios, ya que existen varias librerías que no se encuentran optimizadas y realizan daños al sistema operativo de la Raspberry, las librerías que se han instalado son las más optimas, las cuales no han tenido conflicto en el sistema teniendo un perfecto desempeño, con toda esta información procedimos con la programación de la cámara MLX90640 para que cumpla su función de acuerdo con lo establecido en este proyecto.

3.8 Diagrama de flujo de la programación de la cámara térmica

Figura 3.8.1 Diagrama de flujo de la programación de la cámara térmica



Explicación de la programación de la cámara térmica mediante un diagrama de flujo Fuente: (Villalba. S, 2022)

Como se puede observar en el diagrama de flujo se muestra el proceso que se utilizó para la configuración de la cámara térmica donde la secuencia detecta el objeto y continua con el procedimiento, si no es así finaliza el programa a continuación se explicará con más detalle la codificación del programa.

3.8.1 Programación de la cámara térmica con la Raspberry

Se realizó el llamado de las librerías necesarias para continuar con esta programación.

```
Import time, board,busio
import numpy as np
```

```
import adafruit_mlx90640
import matplot.pyplot as plt
From scipy import ndimage
import argparse
```

se declaró la variable “parser” con sus propios argumentos para poder llamar a la cámara MLX90640.

```
Parser = argparse.ArgumentParser(description=’Thermal Camera Program’)
parser.add_argument(‘—mirror’, dest=’imageMirror’, action=’store_const’,
default=’false’,
                const=’imageMirror’, help=’Flip the image for selfie (default: false)’)
```

aquí aplicamos el modo espejo con la imagen que nos detecte la cámara así funcionando o negando cuando detecte por eso la razón de aplicar un if else.

```
Args = parser.parse_args()
imageMirror = args.imageMirror

if(imageMirror == ‘false’):
   print(‘Mirror mode: false’)
else:
   imageMirror = ‘true’
   print(‘Mirror mode: true’)
```

en esta sección del código lo que realizamos es la configuración de la cámara utilizando la capacidad que posee la cámara MLX90640, pero también se debe aclarar que tanto el HDI a utilizar que sea mayor a 720 fotogramas por segundo limita a este módulo a 2HZ, es mejor utilizar una pantalla propia de la Rasberry que tenga la resolución permitida para este módulo.

```
I2c = busio.I2C(board.SCL, board.SDA, frequency=400000) # setup I2C
mlx = adafruit_mlx90640.MLX90640(i2c) # begin MLX90640 with I2C comm
mlx.refresh_rate = adafruit_mlx90640.RefreshRate.REFRESH_2_HZ
```

```
mlx_shape = (24,32) # mlx90640 shape
mlx_interp_val = 10 # interpolate # on each dimension
mlx_interp_shape = (mlx_shape[0]*mlx_interp_val,
                    mlx_shape[1]*mlx_interp_val)
```

Lo que realizamos en esta sección es el diseño de la interfaz donde se va a observar la imagen captada por la cámara de colores determinados, en este caso se optó a mayor temperatura color rojo y menor temperatura el color azul, siempre se debe tomar en cuenta que se realiza lo más optimo tanto para el módulo como para la Rasberry.

```
Fig = plt.figure(figsize=(12,9)) # start figure
ax = fig.add_subplot(111) # add subplot
fig.subplots_adjust(0.05,0.05,0.95,0.95) # get rid of unnecessary padding
therm1 = ax.imshow(np.zeros(mlx_interp_shape),interpolation='none',
                   cmap=plt.cm.bwr,vmin=25,vmax=45)
cbar = fig.colorbar(therm1) # setup colorbar
cbar.set_label('Temperature [$^{\circ}$C]',fontsize=14)

fig.canvas.draw() # draw figure to copy background
ax_background = fig.canvas.copy_from_bbox(ax.bbox)
ax.text(-75, 125, 'Max:', color='red')
textMaxValue = ax.text(-75, 150, 'test1', color='black')
fig.show() # show the figure before blitting

frame = np.zeros(mlx_shape[0]*mlx_shape[1])
```

Aquí aplica la configuración de la interfaz del medidor de temperatura utilizando las ventajas de nuestra cámara y actualización de la imagen así utilizando la parte de video en vivo, guardando en una variable que luego será utilizada.

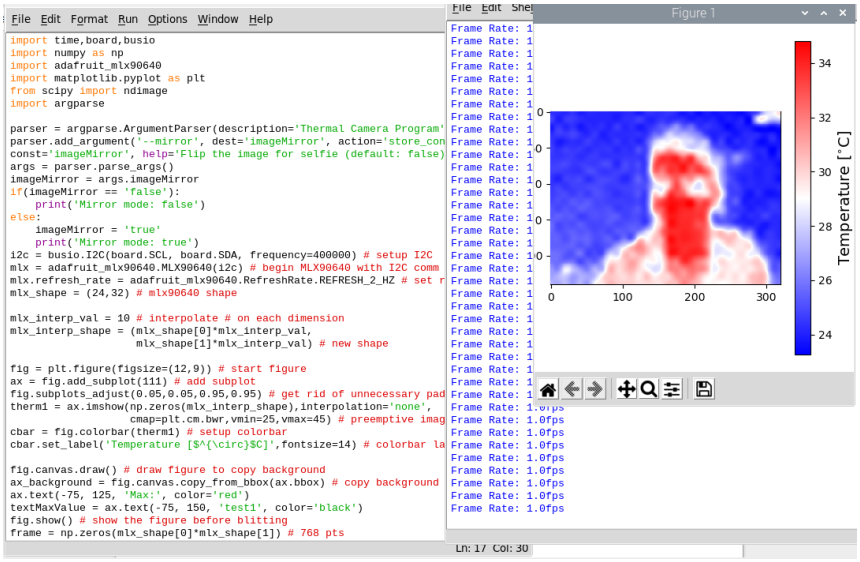
```
Def plot_update():
   fig.canvas.restore_region(ax_background)
   mlx.getFrame(frame) # read mlx90640
   data_array = np.fliplr(np.reshape(frame,mlx_shape))
```

```
    if(imageMirror == 'true'):
        data_array = np.flipud(data_array)
    data_array = ndimage.zoom(data_array,mlx_interp_val) # interpolate
    therm1.set_array(data_array)
    therm1.set_clim(vmin=np.min(data_array),vmax=np.max(data_array))
    cbar.on_mappable_changed(therm1) # update colorbar range
    plt.pause(0.001)
    ax.draw_artist(therm1)
    textMaxValue.set_text(str(np.round(np.max(data_array), 1)))
    fig.canvas.blit(ax.bbox)
    fig.canvas.flush_events()
    fig.show()
    return
```

en esta sección del código lo que realizamos es llamar a nuestra variable la cual programamos con anterioridad para que cumpla su función en la interfaz que se realizó, al correr el código se obtuvo lo siguiente como se muestra en la siguiente figura 3.8.1.1.

```
t_array = []
while True:
    t1 = time.monotonic()
    try:
        plot_update()
    except:
        continue
    t_array.append(time.monotonic()-t1)
    if len(t_array)>10:
        t_array = t_array[1:]
    print('Frame Rate: {0:2.1f}fps'.format(len(t_array)/np.sum(t_array)))
```

Figura 3.8.1.1 Funcionabilidad del programa con la cámara MLX90640



Observación de la Cámara térmica en funcionamiento Fuente: (Villalba. S, 2022)

4. CAPÍTULO 4

En este capítulo indicaremos algunos resultados que se obtuvo al comparar las distintas medidas de dos elementos electrónicos principales de la placa madre, en el cual uno se encontraba en mal estado para poder tener dos tipos de perspectivas al realizar la experimentación con respecto al proyecto.

4.3 Comparación de temperaturas

En esta sección tomamos algunas muestras de algunos dispositivos electrónicos para comparar el porcentaje de error que podría tener nuestro prototipo al medir temperatura como indicamos en las siguientes tablas:

4.1.1 Tabla de valores medidos

Cámara MLX90640	Pirómetro	Error %
48 °C	47 °C	2%
47.5 °C	46 °C	3%
50 °C	48 °C	4%
50.5 °C	50.5 °C	0%
50 °C	49 °C	2%
50.5°C	50 °C	1%
50 °C	49.5 °C	1%
48.5 °C	47.5 °C	2%

Medición de temperatura del procesador para su comparación del prototipo con el termómetro Fuente: (Villalba. S, 2022)

Como se puede observar la comparación de los valores tomados tanto de la cámara como del termómetro no es muy grande lo cual es viable, también tenemos que aclarar que la cámara térmica mide el punto más caliente no generaliza la temperatura en el dispositivo se realizó otras tomas de temperatura con los dispositivos, para que se pueda apreciar esta diferencia como se indica en la siguiente tabla e imágenes.

4.1.2 Tabla de valores medidos chip de video

Cámara MLX90640	Pirómetro	Error %
70 °C	69.6 °C	0.57 %
72.5 °C	72 °C	0.69 %
80 °C	78 °C	2 %
82.5 °C	81 °C	1 %
83 °C	82.6 °C	0.48 %
100 °C	99 °C	1 %
93 °C	92.6 °C	0.43 %
120 °C	119 °C	0.84 %

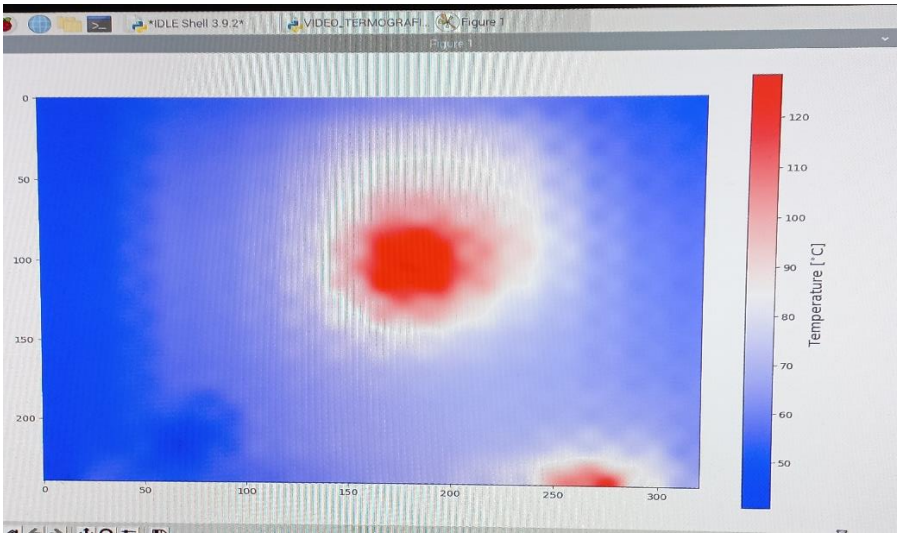
Medición de temperatura en un chip de video cortocircuitado Fuente: (Villalba. S, 2022)

Al comparar las temperaturas en el dispositivo cortocircuitado, la diferencia entre las temperaturas no son tan altas, pero se puede observar que se elevan progresivamente es por el dispositivo que se encuentra en mal estado.

Figura 4.1 Medición con termómetro

Temperatura del chip de video cortocircuitado Fuente: (Villalba. S, 2022)

Figura 4.1.2 Medición con cámara térmica



Chip de video observado con la cámara térmica con su temperatura Fuente: (Villalba. S, 2022)

Como se puede observar en las dos figuras se indica la temperatura que posee el dispositivo dañado, pero existen una gran diferencia la cual nos indica la zona de más calor como indica en la figura 4.1.2 con la cámara térmica que nos da una temperatura máxima de 120°C en la zona del dispositivo dañado, si se observa con atención la temperatura general desde la cámara térmica es de 74°C, en el caso del Pirómetro nos indica una temperatura general de 72°C.

En la aplicación del método clásico al realizar un diagnóstico por el operario es de 2 horas por una placa madre con diversos daños y con un daño mínimo lleva alrededor de 1 hora, al utilizar el método que se realizó con el prototipo el diagnóstico de las placas madre con diversos daños es de 1 hora y con mínimo de daño se reduce a 30 minutos.

4.2 Calculo del error en la comparación de temperaturas

4.2.1 Porcentaje de error

$$\text{ERROR\%} = \left(\frac{|\text{Vexacto} - \text{Vaproximado}|}{Vexacto}\right) 100\% \qquad \text{Ec. (4.2.1)}$$

Donde el valor exacto está representado por el valor tomado del Pirómetro y el valor aproximado está representado por el valor que se obtuvo desde el prototipo.

$$\text{ERROR\%} = \left(\frac{|47\ °\text{C} - 48\ °\text{C}}{48\ °C}\right) \times 100\%$$

$$\text{ERROR\%} = \left(\frac{|-1\ °\text{C}|}{47\ °C}\right) \times 100\%$$

$$\text{ERROR\%} = 2\%$$

Este es el proceso que se realizó para obtener el error en las tablas 4.1 y 4.2, para comparar los valores obtenidos de una mejor manera.

4.3 Calculo de la matriz de confusión

Para calcular la matriz de confusión se necesita los datos que se obtuvieron de la clasificación y los errores que se tuvieron como se muestra en la tabla 4.3.1

4.3.1 Tabla de la matriz de confusión

Procesador	20	0	0
KBC	0	15	1
Bios	1	1	10

Datos de muestras de las neuronas a utilizar Fuente: (Villalba. S, 2022)

Con los datos de la tabla 4.3.1 se puede obtener la matriz de observación donde TP son los verdaderos positivos, TN verdaderos negativo, FP falsos positivos, FN falsos negativos.

4.3.2 Tabla de observación

Matriz de observación				
Clases	Medidas			
	TP	TN	FP	FN
Procesador	20	27	1	0
KBC	15	31	1	1
Bios	10	35	1	2

Con estos datos se podrán obtener la eficiencia y el error de la neurona Fuente: (Villalba. S, 2022)

4.3.1 Calculo de los indicadores

Las ecuaciones que se utilizarán para obtener los valores necesarios son:

$$\textbf{Exacitud} = \left(\frac{\text{TP} + \text{TN}}{TP + TN + FP + FN}\right) \times 100\% \qquad \text{Ec. 4.3.1.1}$$

$$\textbf{Presición} = \left(\frac{\text{TP}}{TP + FP}\right) \times 100\% \qquad \text{Ec. 4.3.1.2}$$

- Procesador

$$\textbf{Exacitud} = \left(\frac{20+27}{20+27+1+0}\right) \times 100\%$$

$$\textbf{Exacitud} = \left(\frac{47}{48}\right) \times 100\%$$

$$\textbf{Exacitud} = 97{,}92\%$$

$$\textbf{Presición} = \left(\frac{20}{20+1}\right) \times 100\%$$

$$\textbf{Presición} = \left(\frac{20}{21}\right) \times 100\%$$

$$\textbf{Presición} = 95{,}24\%$$

- KBC

$$\textbf{Exacitud} = \left(\frac{15+31}{15+31+1+1}\right) \times 100\%$$

$$\textbf{Exacitud} = \left(\frac{46}{48}\right) \times 100\%$$

$$\textbf{Exacitud} = 95{,}83\%$$

$$\textbf{Presición} = \left(\frac{15}{15+1}\right) \times 100\%$$

$$\textbf{Presición} = \left(\frac{20}{21}\right) \times 100\%$$

$$\textbf{Presición} = 93{,}75\%$$

- Bios

$$\textbf{Exacitud} = \left(\frac{10+35}{10+35+1+2}\right) \times 100\%$$

$$\textbf{Exacitud} = \left(\frac{45}{48}\right) \times 100\%$$

$$\textbf{Exacitud} = 93{,}75\%$$

$$\textbf{Presición} = \left(\frac{10}{10+1}\right) \times 100\%$$

$$\textbf{Presición} = \left(\frac{10}{11}\right) \times 100\%$$

$$\textbf{Presición} = 90{,}91\%$$

Como se puede apreciar los datos tanto de la exactitud como de la precisión no son inferiores al 70%.

CONCLUSIONES

Con el fin de desarrollar un dispositivo para el diagnóstico de las placas madre. Se analizaron las distintas tecnologías y funciones de cada módulo a utilizar, se concluye que la termografía, inteligencia artificial son herramientas eficientes porque ayuda en la detección de daños en placas madre para un análisis más profundo de los componentes electrónicos para realizar la reparación correcta del componente que se encuentre fallando.

En el desarrollo del algoritmo se tuvieron algunos problemas con las librerías a utilizar, el motivo fue por el modelo de la Raspberry PI 4 que es la más actual, muchas de las librerías no se encontraban actualizadas lo cual dañaban el sistema operativo sin poder cumplir con la función respectiva al desarrollo, para solucionar este tipo de problema se recurrió a la instalación de librerías independientes que se encuentran mencionadas en el punto 3.2 las cuales abarcaban las funciones necesarias para realizar nuestro proyecto sin ningún error.

Dentro de la instalación del módulo de la cámara térmica MLX90640 se tuvo el problema con la instalación de los drivers de esta cámara, se solucionó únicamente con la instalación básica de los drivers y la utilización del IDE Python con el que es compatible, la resolución de la cámara es configurable al máximo de fotogramas de 16 HZ que nos entrega la misma, hay que aclarar que la resolución del HMI puede causar caída de fotogramas en la imagen que entregué la cámara.

En cuanto en las pruebas que se realizó en el punto 4.1 del dispositivo en placas madre se concluye que es un 96% aceptable en cuanto a exactitud, ya que las 16 muestras tomadas de dos placas madre diferentes, al comparar las temperaturas tomadas con el dispositivo desarrollado y un pirómetro se obtuvo un error máximo del 4% en una de las medidas de temperatura en la tabla 4.1 donde se observó la diferencia entre ambos dispositivos, en el proyecto al ser utilizado tenemos medidas parciales y no medidas generales como en el pirómetro.

En una de las pruebas de temperaturas las medidas eran muy variables que se observan en la figura 4.1 y 4.2 donde se puede realizar claramente una comparación de las medidas, se concluyó que al momento de las mediciones con los dos dispositivos especialmente en la placa madre con daños se puede observar que un componente electrónico en mal estado

ocasiona elevaciones de temperaturas altas hasta llegar al punto de deformar el componente electrónico (chip video) y provocar un daño severo a la placa madre.

RECOMENDACIONES

Para un futuro se puede realizar cambios en el diseño del dispositivo como añadir le una fuente independiente y así convirtiendo en portátil así dando una mejor experiencia al usuario.

No se recomienda utilizar cualquier tipo de HMI por la resolución de la cámara térmica, en lugar de ello mejor es utilizar un módulo HMI propio de la Rasberry para que la imagen de la cámara del 100% de su rendimiento.

Se recomienda utilizar las herramientas propias de la Rasberry PI 4 y librerías recomendadas las cuales se encuentran actualizadas de acuerdo con el modelo de tarjeta que se utilice.

REFERENCIAS BIBLIOGRÁFICAS

Basogain.X,(2018), *Redes neuronales artificiales y sus aplicaciones*,Pp.4-6, Disponible en: https://ocw.ehu.eus/pluginfile.php/40137/mod_resource/content/1/redes_neuro/contenidos/pdf/libro-del-curso.pdf

Conceptos Definición (derechos reservados). (2022), *Procesador*, Disponible en: https://conceptodefinicion.de/procesador/

Datasheet café. (19 de abril 2022), *KB9012 Datasheet – Keyboard Controller*, Disponible en: http://www.datasheetcafe.com/kb9012-datasheet-keyboard-controller/

Genius, (2019), *520498 CAMARA WEB C/MICROFONO 1000X 720P*, Disponible en: GENIUSfile:///D:/Downloads/CL_520498_FT.pdf

HP. (2022), *¿Qué es el BIOS? Funciones básicas y cómo acceder a la configuración de Windows*, Disponible en: https://www.hp.com/mx-es/shop/tech-takes/como-acceder-a-la-configuracion-del-bios-en-una-pc-con-windows#:~:text=Como%20el%20programa%20de%20inicio,responsable%20de%20iniciar%20tu%20sistema.

Hussein, M.; Mutlag, A. (2019). *Face Detection Methods a Comparative Study between Viola-Jones and Skin Color Detection. Journal of Engineering and Applied Sciencies*, 14 (14). Disponible en: https://www.researchgate.net/publication/338080735_Face_Detection_Methods_A_Comparative_Study_Between_Viola-Jones_and_Skin_Color_Detection

Jeremias.A,(2022), *Reconocimiento de objetos a traves de la metedologia Haar Cascade*,Pp.4-5, Disponoble en: https://confedi.org.ar/wp-content/uploads/2020/12/Articulo1-RADI16.pdf

Leticia,C.(2022,10 de marzo), *¿Que es una raspberry y para qué sirve?*. https://es.godaddy.com/blog/que-es-raspberry-pi/#:~:text=Una%20Raspberry%20PI%20es%20un,RAM%20de%20hasta%208%20GB.

Melexis. (2019,2 de febrero), *Cámara térmica MLX90640*. Disponible en: https://html.alldatasheet.com/html-pdf/1149434/MELEXIS/MLX90641/18543/53/MLX90641.html

Nita, L.; Peña, E. (2019). *Principios Básicos de la termografía infrarroja y su utilización como técnica para mantenimiento predictivo*. pp.56-63. Disponible en: https://repository.upb.edu.co/bitstream/handle/20.500.11912/1561/digital_20999.pdf?sequence=1&isAllowed=y

Rouhiainen.L, (2018). *Inteligencia artificial 101 cosas que debes saber hoy sobre nuestro futuro, Primera edicion noviembre 2018*, Editorial de centros de libros PAPF,SLU Pp.16-20. Disponible en: https://www.planetadelibros.com/libros_contenido_extra/40/39307_Inteligencia_artificial.pdf

Sánchez, S.; Granados, A. (2019). *La innovación y tecnología como estrategias en las empresas del sector comercio del distrito de Santa Marta, Universidad cooperativa de Colombia Facultad de ciencias administrativa, contables y comercio internacional, Colombia*, pp 21-23. Disponible en: https://repository.ucc.edu.co/bitstream/20.500.12494/16918/2/2019_innovaci%C3%B3n_tecnolog%C3%ADa_estrategias.pdf

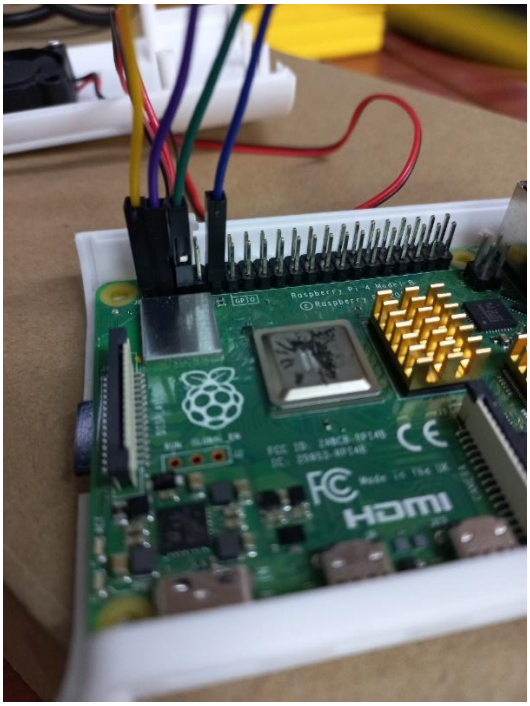
Sa-Nguannarm, P., Charoenpong, T., Chianrabutra, C., & Kiatsoontorn,K.(2019). *A Method of 3D Hand Movement Recognition by a Leap Motion Sensor for Controlling Medical Image in an Operating Room. 2019 1st International Symposium on Instrumentation, Control, Artificial Intelligence,and Robotics*, ICA-SYMP 2019,17–20. Disponible en : https://doi.org/10.1109/ICA-SYMP.2019.8645985

ANEXOS

ANEXO 1. Armado y instalación del módulo de cámara térmica

Anexo 1.1 Intalación de la Rasberry en su carcasa de protección

Anexo 1.2 instalación de la cámara térmica con la tarjeta Rasberry



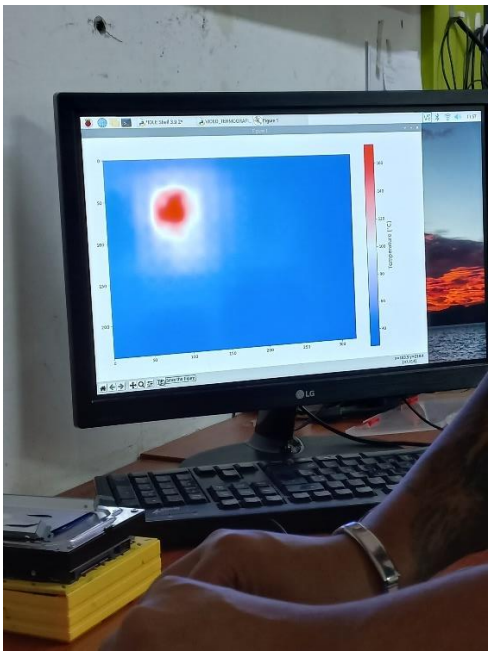
ANEXO 2. Cámaras a utilizar para el proyecto

Anexo 2.1 Módulo de la cámara Térmica MLX90640

Anexo 2.2 Cámara Genius 1000x

ANEXO 3. Prueba con placa madre que contiene daños

Anexo 3.1 Se muestra el daño interno del dispositivo electronico con mayor intensidad de color rojo



Anexo 3.2 La medida de temperatura del dispositivo es mínima la razon es porque se lo realizó con una corriente menor a la del encendido el motivo fue porque al calentarse el dispositivo cortocircuitado se deformaba en segundos por el calor que emitia para poder tener una comparación con nuestro prototipo.