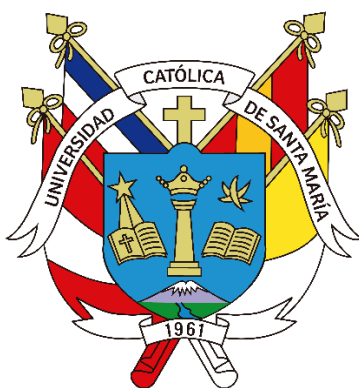


Universidad Católica de Santa María
Facultad de Ciencias e Ingenierías Físicas y Formales
Escuela Profesional de Ingeniería Electrónica



**Implementación de un Sistema de Detección de Equipos de Protección
Personal Mediante Visión Artificial para Trabajadores del Sector Industrial**

Tesis presentada por el Bachiller:
Oviedo Agramonte, Danny Arnaldo
ORCID: 0009-0005-5369-2164

para optar el Título Profesional de Ingeniero Electrónica
con Especialidad en Automatización y Control

Asesor (a):
Ing. Málaga Chávez, César Eduardo
ORCID: 0000-0002-5461-5380

Arequipa – Perú
2024

UCSM-ERP

UNIVERSIDAD CATÓLICA DE SANTA MARÍA
INGENIERIA ELECTRONICA
CON ESPECIALIDAD EN AUTOMATIZACIÓN Y CONTROL
TITULACIÓN CON TESIS
DICTAMEN APROBACIÓN DE BORRADOR

Arequipa, 18 de Enero del 2024

Dictamen: 003807-C-EPIE-2024

Visto el borrador del expediente 003807, presentado por:

2008240491 - OVIEDO AGRAMONTE DANNY ARNALDO

Titulado:

**IMPLEMENTACIÓN DE UN SISTEMA DE DETECCIÓN DE EQUIPOS DE PROTECCIÓN PERSONAL
MEDIANTE VISIÓN ARTIFICIAL PARA TRABAJADORES DEL SECTOR INDUSTRIAL**

Nuestro dictamen es:

APROBADO

**29267682 - RODRIGUEZ GONZALES PEDRO ALEX
DICTAMINADOR**



**29329443 - VALDIVIESO HERRERA DIANA ISABEL
DICTAMINADOR**



**29410027 - SULLA TORRES RAUL RICARDO
DICTAMINADOR**



Implementación de un Sistema de Detección de Equipos de Protección Personal Mediante Visión Artificial para Trabajadores del Sector Industrial

ORIGINALITY REPORT

14%

SIMILARITY INDEX

13%

INTERNET SOURCES

0%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1	repositorio.ug.edu.ec Internet Source	5%
2	omes-va.com Internet Source	3%
3	ciberninjas.com Internet Source	1%
4	dspace.unl.edu.ec Internet Source	1%
5	Submitted to Universidad Católica de Santa María Student Paper	1%
6	es.scribd.com Internet Source	1%
7	es.wikipedia.org Internet Source	1%
8	Submitted to Universidad Nacional de Educación a Distancia Student Paper	1%

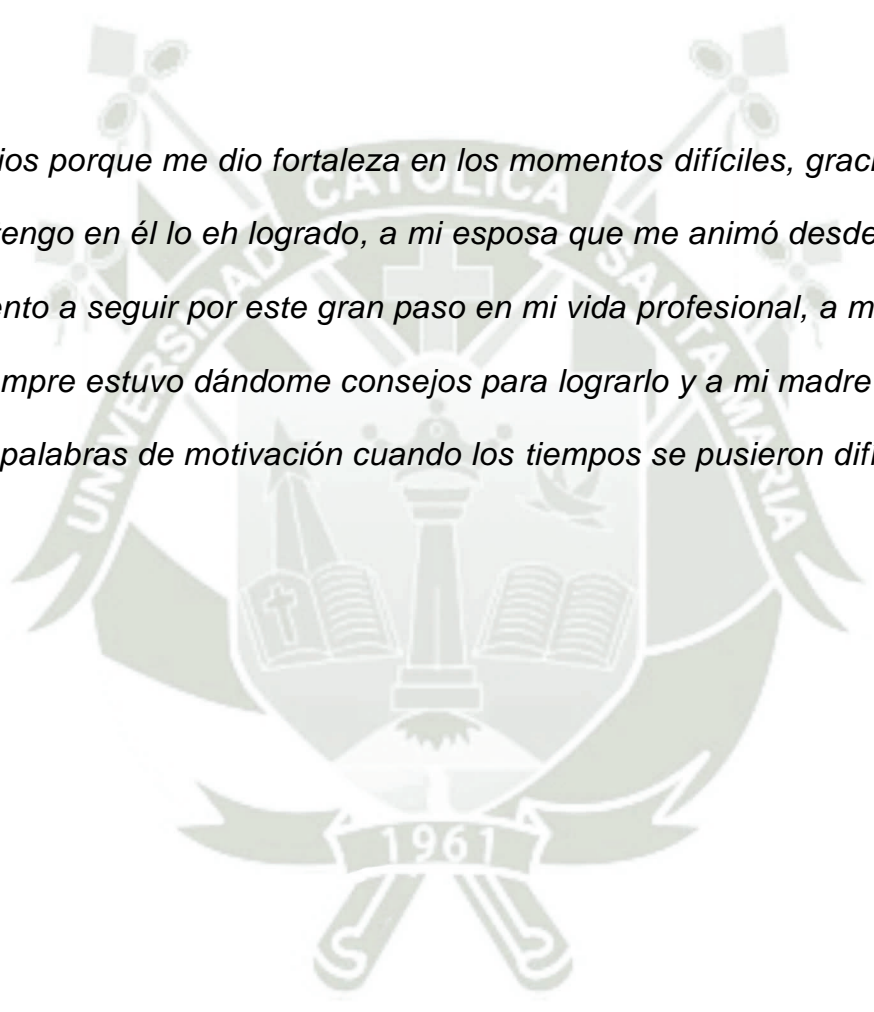


DEDICATORIA

A mi querida hija que al escuchar su voz siempre me llena de alegría, siempre será la luz de mi existir, al pensar en ella todo es posible porque ilumina mi camino, me da fuerzas para seguir adelante en cada paso que doy y su presencia es la energía que necesito cada día.

AGRADECIMIENTO

A Dios porque me dio fortaleza en los momentos difíciles, gracias a la fe que tengo en él lo eh logrado, a mi esposa que me animó desde un primer momento a seguir por este gran paso en mi vida profesional, a mi padre que siempre estuvo dándome consejos para lograrlo y a mi madre con sus palabras de motivación cuando los tiempos se pusieron difíciles.



RESUMEN

La investigación se presenta con el objetivo de implementar un sistema de detección de equipos de protección personal (EPP) mediante visión artificial y aprendizaje de máquina para supervisar a los trabajadores del sector industrial. La metodología de investigación incluyó una investigación bibliográfica y hemerográfica para obtener información relevante sobre el tema, así como técnicas e instrumentos para el desarrollo experimental.

La investigación abordó la importancia de la implementación de sistemas de seguridad en el lugar de trabajo y la legislación relacionada con el uso de EPP.

El estudio se llevó a cabo en una habitación simulada que representaba un área de trabajo industrial, utilizando tecnologías y métodos actuales de visión artificial para la detección de EPP en tiempo real. Se emplearon técnicas de análisis y respuesta del algoritmo de aprendizaje de máquina a la toma de fotografías o video por las cámaras, y se evaluaron las pruebas con el cumplimiento de uso de los EPP en los trabajadores.

Las conclusiones, destacaron la importancia de implementar sistemas de detección de EPP para reducir los accidentes laborales, ya que el incumplimiento del uso de EPP suele ser una de las causas de inestabilidad laboral, pérdidas económicas y desconfianza en la contratación de nuevos empleados. Se mencionó que la tecnología de visión artificial y aprendizaje de máquina tiene el potencial de mejorar significativamente la seguridad laboral y la conformidad con las normativas de seguridad, contribuyendo a la disminución de posibles accidentes, incidentes y lesiones en el lugar de trabajo.

Palabras Claves: Visión Artificial, Aprendizaje de Maquina, Seguridad Laboral

ABSTRACT

The research is presented with the objective of implementing a personal protective equipment (PPE) detection system using machine vision and machine learning to monitor workers in the industrial sector. The research methodology included a bibliographic and hemerographic research to obtain relevant information on the subject, as well as techniques and instruments for experimental development.

The research addressed the importance of the implementation of safety systems in the workplace and the legislation related to the use of PPE.

The study was conducted in a simulated room representing an industrial work area, using current machine vision technologies and methods for real-time PPE detection. Machine learning algorithm analysis and response techniques were employed in response to the taking of photographs or video by the cameras, and the evidence was evaluated with the workers' compliance in the use of PPE.

The conclusions highlighted the importance of implementing PPE detection systems to reduce workplace accidents, since non-compliance with PPE use is often one of the causes of job instability, economic losses and distrust in hiring new employees. It was mentioned that machine vision and machine learning technology has the potential to significantly improve occupational safety and compliance with safety regulations, contributing to the reduction of potential accidents, incidents and injuries in the workplace.

Keywords: Machine Vision, Machine Learning, Occupational Safety, Workplace Safety.

ÍNDICE

DEDICATORIA.....	iii
AGRADECIMIENTO.....	iv
RESUMEN	v
ABSTRACT	vi
ÍNDICE	vii
ÍNDICE DE TABLAS.....	xii
ÍNDICE DE FIGURAS.....	xiii
INTRODUCCIÓN	1
CAPITULO I: GENERALIDADES.....	2
1.1 IDENTIFICACIÓN DEL PROBLEMA	2
1.2 DESCRIPCIÓN DEL PROBLEMA	3
1.3 OBJETIVOS	5
1.3.1 Objetivos generales.....	5
1.3.2 Objetivos específicos.....	5
1.4 VARIABLES	6
1.4.1 Variables independientes	6
1.4.2 Variables dependientes	6
1.4.3 Indicadores de variables.....	6
1.5 HIPÓTESIS	6
1.6 ALCANCES	7

1.7	LIMITACIONES	7
2	CAPITULO II: MARCO TEÓRICO.....	9
2.1	ANTECEDENTES DE LA INVESTIGACIÓN	9
2.1.1	Antecedentes internacionales.....	9
2.1.2	Antecedentes Nacionales.....	11
2.1.3	Antecedentes locales.....	13
2.2	BASES TEÓRICAS	15
2.2.1	Equipos de protección personal.....	15
2.2.2	Visión artificial.....	21
2.2.3	Cámaras.....	29
2.2.4	Detección de forma y color.....	34
2.2.5	Reconocimiento de patrones.....	37
2.2.6	Algoritmo LBPH.....	37
2.2.7	Ordenador de programación.....	38
2.2.8	Microcontroladores Raspberry Pi OS.....	39
2.2.9	Algoritmos.....	46
2.2.10	Lenguajes de programación.....	51
2.2.11	Librerías de programación.....	52
2.2.12	Kali Linux.....	55
2.2.13	Ubuntu MATE.....	56
2.2.14	RISC OS.....	57

2.3 ESQUEMA CONCEPTUAL 58

3 CAPITULO III: ASPECTOS GENERALES, METODOLOGÍA Y

DESARROLLO..... 61

3.1 ASPECTOS METODOLÓGICOS 61

3.1.1 Estrategia 61

3.1.2 Tipo de investigación 61

3.1.3 Nivel de investigación 61

3.1.4 Métodos 62

3.2 CAMPO DE VERIFICACIÓN 64

3.2.1 Ubicación espacial:..... 64

3.2.2 Ubicación Temporal:..... 64

3.2.3 Unidades de Estudio:..... 64

3.3 TÉCNICAS E INSTRUMENTOS 65

3.3.1 Técnicas 65

3.3.2 Instrumentos 65

3.4 ASPECTOS GENERALES DE INSTRUMENTOS UTILIZADOS

66

3.4.1 RASPBERRY PI 3 67

3.4.2 Hardware Broadcom BCM2837 72

3.4.3 USB Chip 73

3.4.4 GPIO..... 74

3.4.5	Antena	75
3.4.6	Memorias.....	76
3.4.7	Conector de alimentación	77
3.4.8	Conector HDMI	78
3.4.9	Conector de audio	78
3.4.10	Conectores Ethernet y USB	79
3.4.11	Conectores de cámara y display	80
3.4.12	Software	81
3.4.13	Raspbian	81
3.5	DESARROLLO EXPERIMENTAL	82
3.5.1	Diseño de proceso de interconexión entre los diferentes componentes.....	83
3.5.2	Conexión de alimentación de Raspberry Pi.....	83
3.5.3	Encendido del prototipo	85
3.5.4	Configuración inicial del sistema operativo	86
3.5.5	Configuración inicial de Raspbian.....	86
3.5.6	THONNY PHYTON IDE.....	88
3.5.7	PROGRAMACION Y EJECUCION.....	91
4	CAPITULO IV: RESULTADOS.....	104
5	CAPITULO V: PRESUPUESTO	109
5.1	Coste de Hardware.	109

5.2	Coste de Software	109
5.3	Coste de Personal	110
5.4	Gasto global	110
CONCLUSIONES.....		112
RECOMENDACIONES		112
REFERENCIA		117
ANEXOS		121



ÍNDICE DE TABLAS

Tabla 1 <i>Cantidad de Cámaras</i>	65
Tabla 2 <i>Cantidad de Ordenadores</i>	66
Tabla 3 <i>Cantidad de Epp</i>	66
Tabla 4 <i>Especificaciones De Raspberry Pi 3</i>	71
Tabla 5 <i>Conectores De Raspberry Pi 3</i>	71
Tabla 6 <i>Designaciones De Clase De Tarjetas SD</i>	77
Tabla 7 <i>Test</i>	107
Tabla 8 <i>Costos del hardware</i>	109
Tabla 9 <i>Costos del software</i>	109
Tabla 10 <i>Costos del personal</i>	110
Tabla 11 <i>Costos de los EPP</i>	110
Tabla 12 <i>Resumen del presupuesto total.</i>	111
Tabla 13 <i>Cronograma</i>	¡Error! Marcador no definido.

ÍNDICE DE FIGURAS

Figura 1 <i>Caja Negra</i>	5
Figura 2 <i>Partes constitutivas de los cascos</i>	16
Figura 3 <i>Protección de ojos</i>	17
Figura 4 <i>Protección de rostro</i>	17
Figura 5 <i>Etapas fundamentales para el sistema de visión artificial</i>	23
Figura 6 <i>Imagen Original</i>	24
Figura 7 <i>Imagen procesada de un objeto</i>	24
Figura 8 <i>Segmentación nuclear de células</i>	25
Figura 9 <i>Construcción lineal de imagen</i>	31
Figura 10 <i>Obtención de la imagen mediante barridos sucesivos</i>	31
Figura 11 <i>Cámara web o webcams</i>	34
Figura 12 <i>Interfaz de Raspberry Pi OS</i>	40
Figura 13 <i>Logo Raspberry Pi Foundation</i>	42
Figura 14 <i>Componentes De Una Raspberry Pi</i>	43
Figura 15 <i>Modelos De Raspberry Pi (A)</i>	44
Figura 16 <i>Modelos De Raspberry Pi (B)</i>	45
Figura 17 <i>Matriz de distancias</i>	48
Figura 18 <i>Resultados del diagrama de grupo de asociación simple y completo</i>	48
Figura 19 <i>Kali Linux</i>	56
Figura 20 <i>Escritorio de UbuntuMATE</i>	57
Figura 21 <i>Interfaz de escritorio</i>	58
Figura 22 <i>Diagrama de funcionamiento del detector de EPP</i>	59
Figura 23 <i>Raspberry Pi 3 Modelo B</i>	67

Figura 24 Componentes De Una Raspberry Pi 3	69
Figura 25 Esquemático Raspberry Pi 3 (B)	70
Figura 26 Broadcom Bcm2837.....	73
Figura 27 Usb Chip.....	74
Figura 28 Gpio.....	75
Figura 29 Antena.....	76
Figura 30 Conector De Alimentación.....	77
Figura 31 Conector HDMI.....	78
Figura 32 Conector De Audio	79
Figura 33 Conectores Ethernet Y Usb.....	80
Figura 34 Conectores De Cámara Y Display.....	81
Figura 35 Escritorio Raspbian Jessie	82
Figura 36 Diseño De Interconexión De Componentes	83
Figura 37 Inserción De Memoria SD En Raspberry Pi	84
Figura 38 Conversor VGA/HDMI o HDMI/HDMI	85
Figura 39 Acoplamiento Y Conexión De Raspberry Pi En Monitor.....	85
Figura 40 Actualización Cache De Apt Raspbian	87
Figura 41 Actualización De Todo El Software De Raspbian.....	88
Figura 42 Thonny Phyton.....	91
Figura 43 Entorno de Programación de Generador de Base de Datos	92
Figura 44 Entorno de Programación para entrenar modelos.....	95
Figura 45 Archivo XML generado por el modelo entrenado	96
Figura 46 Entorno de Programación Detección de Mascarilla.....	97
Figura 47 Entorno de Programación Detección Múltiple	97
Figura 48 Diagrama de flujo del reconocimiento en tiempo real.....	102

Figura 49 <i>Matriz de confusión</i>	104
Figura 50 <i>Entorno de Programación “Detección de Mascarilla”</i>	105
Figura 51 <i>Entorno de Programación “Detección de Lentes”</i>	105
Figura 52 <i>Entorno de Programación “Detección de Casco”</i>	105
Figura 53 <i>Entorno de Programación “Detección de Casco”</i>	106
Figura 54 <i>Entorno de Programación “Detección de Lentes”</i>	106



INTRODUCCIÓN

La presente investigación aborda un problema significativo en el ámbito laboral, específicamente en el sector industrial, donde los accidentes laborales pueden tener consecuencias graves tanto para los trabajadores como para las empresas. La implementación de un sistema de detección de EPP mediante visión artificial y aprendizaje de máquina se presenta como una solución innovadora que tiene el potencial de reducir significativamente la incidencia de accidentes laborales al garantizar el cumplimiento del uso de equipos de protección personal.

En el primer capítulo, se identifica el problema de los accidentes laborales en el sector industrial y se establecen los objetivos, variables, hipótesis, alcances y limitaciones de la investigación. En el segundo capítulo, se presenta el marco teórico, que incluye antecedentes internacionales, nacionales y locales, así como las bases teóricas de los equipos de protección personal, la visión artificial, las cámaras, la detección de forma y color, el reconocimiento de patrones, los algoritmos, los lenguajes y las librerías de programación, entre otros. En el tercer capítulo, se describen los aspectos generales, la metodología y el desarrollo experimental de la investigación. En el cuarto capítulo, se presentan los resultados obtenidos a partir de la implementación del sistema de detección de EPP mediante visión artificial y aprendizaje de máquina. Finalmente, en el quinto capítulo, se presenta el presupuesto del proyecto, así como las conclusiones y recomendaciones derivadas de la investigación.

CAPITULO I: GENERALIDADES

1.1 IDENTIFICACIÓN DEL PROBLEMA

Muchos de los trabajadores industriales, obras civiles, construcción entre otros, se encuentran sometidos a sufrir algún inconveniente que puede poner en peligro su salud, el cual afecta la actividad laboral de la empresa generando gastos en la recuperación del estado de salud o hasta indemnización por la muerte del personal de trabajo (School, 2019).

Por ende, el Ministerio de Trabajo y Promoción de Empleo (MTPE) manifiesta que un accidente es aquella acción que, debido a algunos factores, ya sean personales o del ambiente laboral (orden, atención, higiene, comunicación, etc.), que producen daños en persona o directamente en el ámbito estructural.

Hoy en día, el Perú afronta este tipo de acontecimientos que son cada vez más concurrentes. En el último informe redactado por (Ministerio de Trabajo y Promoción del Empleo, 2016) menciona que en el año 2016, se llevaron a cabo alrededor de 21 mil accidentes laborales, presentando a Lima Metropolitana como la zona donde se evidencio mayor cantidad de estos acontecimientos. De acuerdo al tipo de trabajo (actividad económica) se relaciona el tipo de accidente que se puede sufrir, siendo unos más peligrosos que otros; siendo la industria uno de los trabajos con mayor frecuencia de incidentes y /o accidentes. (Conexionesan, 2018).

Según la ley 29783 y en el artículo 61, el empleador debe adecuar el ambiente de trabajo, con las condiciones e implementos necesarios que se requieran tanto para laborar, como para asegurar la integridad de los trabajadores. Y en el Decreto Legislativo 5-2012-TR artículo 97, señala el uso de

la indumentaria para la protección de los trabajadores (brindar y supervisar el uso de EEP), además a lo estipulado en el artículo 60 de la Ley, estos implementos deben cumplir con el tamaño y las medidas adecuadas que posee el trabajador.

Existen resoluciones que implementan políticas que favorecen al cumplimiento adecuado de los lineamientos en relación al área de salud ocupacional; en otras palabras, son acciones enfocadas a velar por la salud, física y mental de los empleados. Estas se logran y reflejan a través de revisiones médicas, campañas de salud e información, charlas de seguridad, seguro médico, eliminación de posibles peligros, controles diarios de EPP, etc.

Toda empresa posee una política de trabajo en el ámbito de salud y seguridad ocupacional, resguardando la seguridad de los trabajadores, como de los encargados que dirigen, actuando de manera profesional y eficiente.

1.2 DESCRIPCIÓN DEL PROBLEMA

Generalmente los accidentes laborales en una empresa, fábrica u otros ocurre por el incumplimiento del empleo del equipo de seguridad personal, llamado también EPP. Y estos generan inestabilidad laboral, los cuales están en torno a:

- Pérdidas económicas que no solo afectan a la empresa, sino a los familiares del trabajador o empleado.
- Empresas donde ocurre accidentes, y esto genera desconfianza en la contratación de nuevos empleados.
- La pérdida de producción.

Se dice que, en el momento de lidiar con posibles accidentes, incidentes de riesgo, se debe aplicar medidas y/o técnicas que puedan resguardar nuestra seguridad, o la posibilidad de reducir la intensidad del posible accidente. Por ello existen sistemas que ayudan a la reducción de accidentes, con las organizaciones que lleven a mejorar permanentemente las condiciones de los elementos, productos o herramientas necesarias, como los servicios que se disponen, por ende se debe tener un control que permita determinar estas condiciones, así como proponer programas que ayuden a cumplir con los objetivos y la política de la empresa. (NUEVAS ISO, 2020).

El desarrollo de un nuevo sistema de gestión colabora con el progreso, sostenimiento y beneficio de distintos sistemas de gestión. La Plataforma Tecnológica que estos sistemas normalizados como los modelos de excelencia permitan relacionarse o complementarse. Por ende, la aplicación del sistema de visión ficticia, es decir artificial, está siendo usado en muchos artículos, proyectos e investigaciones como una herramienta que ayude a mitigar o prevenir accidentes.

La tecnología de reconocimiento de imágenes, a través del SVA de qué forma ayudaría al reconocimiento y el cumplimiento de los EPP, dentro de una empresa u otra organización que requiera seguridad y disciplina de los trabajadores.

Y **actualmente** la Región de Arequipa ha manifestado un gran número de accidentes laborales, ocupando el segundo lugar a nivel nacional., superado por Lima con 3068 casos durante un año (Rodríguez, 2017).

Para reducir accidentes **se desea** aplicar un Sistema de Detección donde jugaría un papel fundamental la Visión Artificial conjuntamente con el Aprendizaje Máquina, por medio del procesamiento digital de imágenes y la electrónica inalámbrica. Así, dentro de una empresa se realizaría la supervisión y cumplimiento de los EPP.

Figura 1

Caja Negra



Nota. Reconocimiento de la Situación actual y la Situación deseada Elaborado en (diaw.exe 0.97.2)

1.3 OBJETIVOS

1.3.1 Objetivos generales

- Implementar y proponer un sistema de detección de equipos de protección personal (EPP) mediante ciencia aplicada de visión artificial y máquina de aprendizaje para supervisión de trabajadores del sector industrial.

1.3.2 Objetivos específicos

- Identificar métodos de Visión Artificial.
- Identificar y extraer algoritmos de procesamiento de imágenes.
- Instituir algoritmos en lenguaje de programación para la detección de EPP.
- Evaluar los algoritmos y técnicas de visión artificial

- Identificar los equipos a procesar para verificar la complejidad y las técnicas que se deben aplicar para procesar.

Obtener un sistema que permita realizar la inspección de Equipos de protección personal con una perspectiva a integrar más elementos al mismo sistema.

1.4 VARIABLES

1.4.1 Variables independientes

- Trabajadores del sector industrial.

1.4.2 Variables dependientes

- Sistema de detección.

1.4.3 Indicadores de variables

Uso de los elementos de resguardo personal

- Reconocimiento de Mascarilla
- Reconocimiento de Lentes
- Reconocimiento de Casco
- Reconocimiento de Casco y Lentes

Adicionalmente se plantea tener un Generador de Base de Datos, que permita tener a futuro, la integración de más elementos al sistema.

1.5 HIPÓTESIS

Es posible que con activación de un sistema de detección de equipos de protección personal EPP, se podría reducir los accidentes dentro de un ambiente laboral, ya que este sistema de visión artificial cuenta con la capacidad de

distinguir con precisión si los trabajadores están utilizando estos tres tipos específicos de EPP (casco, gafas, mascarilla) o si no los están utilizando.

Además, esto ayuda a la supervisión y la obligación del cumplimiento del uso de los EPP dentro de un área de trabajo, mejorando significativamente la seguridad laboral y la conformidad con las normativas de seguridad, contribuyendo así a la disminución de posibles accidentes, incidentes y lesiones que se pueden dar en el lugar de trabajo.

1.6 ALCANCES

El presente proyecto aplicará la tecnología de Sistema de Visión Artificial (SVA) para la detección de EPP dentro de una habitación, suponiendo que este fuera un área de trabajo industrial.

Con la ayuda del procesamiento digital de imágenes, aprendizaje profundo y otras herramientas, se tratará de identificar los equipos de protección del trabajador, es decir ver si una persona está usando o no los EPP, a través del algoritmo que se desarrollará se podrá indicar, por medio de alertas en cuadros y etiquetas.

Para desarrollar el proyecto se usará los módulos de cámara web y un computador de bajo costo con un tamaño compacto (Raspberry Pi) como parte del hardware. Tomando en cuenta que este tipo de proyectos requieren un desarrollo de algoritmos de visión artificial, procesamiento de video y aprendizaje profundo usaremos el lenguaje de programación de Python.

1.7 LIMITACIONES

El proyecto se limita en realizarse las pruebas dentro de una empresa industrial, tampoco se podrá identificar todos los EPP, por la complejidad del

proyecto y el procesamiento que se requiere para este tipo de tecnología. En esta tesis se busca realizar las pruebas con equipos de protección personal visibles y que se encuentre cerca al rostro como son la mascarilla, el casco y las gafas.

Se limita en la aplicación de los conocimientos sean tan robusto, tal es el caso del enfoque, iluminación, procesamiento rápido de la toma de fotografías.



CAPITULO II: MARCO TEÓRICO

2.1 ANTECEDENTES DE LA INVESTIGACIÓN

2.1.1 Antecedentes internacionales

Massiris et al., (2018) propone en su investigación el uso de una red neuronal convolucional para detectar los EPP. Cuyo objetivo principal fue formular una propuesta tecnológica centrada en la visión artificial para así medir cuantitativa el uso de EPP, con una red neuronal convolucional (CNN) de tipo Yolo, “You Only Look Once”, para el reconocimiento de cascos, guantes, ropa de alta visibilidad de los obreros. Esta investigación es de tipo experimental donde se presentan experimentos, mientras que los productos de esta indagación buscan obtener un eje de estudio del caso y plantear posibles futuros trabajos. La población que se estudió en esta investigación fueron trabajadores metalúrgicos, y se utilizaron las herramientas como el Python para evaluar los resultados y experimentos además del YOLOv2, Keras, OpenCV. Además, se puede determinar la efectividad del sistema YOLO gracias a sus parámetros de calidad y velocidad. Tiene como una de las conclusiones que al realizarse un análisis de caso los resultados revelan que el detector YOLO utilizado era útil en varias áreas con sus respectivas características, comprobándose que son relativamente eficientes en tiempos procesados. Los resultados se pueden mejorar complementando con un supresor de no máximos para así disminuir los etiquetados múltiples.

Karlsson y Strand (2022) Con el propósito de su proyecto fue mejorar la seguridad en el lugar de trabajo mediante el procesamiento de una técnica que

utiliza visión por computadora para reconocer el uso adecuado del equipo de seguridad, aumentando así la seguridad de los trabajadores. La metodología incluyó el reconocimiento de objetos, rostros y personas, utilizando código Python con bibliotecas como OpenCV, face_recognition y CVZone. Los resultados revelaron una precisión promedio del 99% a una distancia de 3 metros y del 89% a 5 metros, con un mejor rendimiento en distancias más cortas y en entornos menos distractores. El sistema detectó de manera confiable cascos y chalecos de seguridad, pero enfrentó desafíos con objetos más pequeños y cuando las características faciales estaban oscurecidas. Aunque la velocidad de procesamiento de la CPU no fue óptima, su implementación en una GPU puede alcanzar la velocidad deseada. En conclusión, el sistema es efectivo dentro de su entorno diseñado, especialmente en escenarios de salas de aire, pero el reconocimiento facial puede requerir un mayor desarrollo, explorando posiblemente códigos QR u otros algoritmos alternativos.

Lopez y Gonzales (2023) En su estudio aborda el aumento de accidentes laborales en los últimos años, vinculándolo a una supervisión deficiente del empleo de equipos de protección personal (EPP). Se propone desarrollar un sistema que emplea inteligencia artificial (IA) a través de la red neuronal YOLO para identificar el uso de EPP con alta precisión. Los resultados muestran que el reconocimiento automático de EPP es viable, con la única salvedad de los guantes, que pueden dar falsos positivos, solucionables mediante un sistema redundante. Además, se creó un conjunto de datos personalizado para entrenar la red, logrando una precisión del 98.13% y un recall del 86.78%. En conclusión, el sistema basado en YOLOv5s es adecuado para aplicaciones en tiempo real y puede mejorar aún más con un equilibrio de clases en el conjunto de datos, lo

que beneficiaría especialmente la identificación de objetos desafiantes. La utilización de una fuente de video o cámara web con redes neuronales convolucionales aporta ventajas, al permitir el reconocimiento temporal en secuencias de video y mejorar la precisión en entornos dinámicos. El aumento de datos también es crucial para generalizar y reducir el sobreajuste, resultando en modelos de visión por computadora más robustos y precisos

2.1.2 Antecedentes Nacionales

Purizaca, (2019) En su tesis “Diseño de laboratorio para pruebas de aislamiento eléctrico a equipos de protección personal para garantizar los trabajos según norma ISO/ IEC 17025:2006 – Chiclayo”. Cuyo objetivo principal fue disminuir los accidentes eléctricos relacionados a Empresas Eléctricas de Chiclayo y optimizar las condiciones en las que deben estar los equipos de resguardo personal, instituyendo un protocolo del mismo. Siendo una indagación de tipo no experimental, analítico y descriptivo, para la adquisición de data se utilizaron técnicas e instrumentos tales como guías de exploración, encuestas y entrevistas, realización de ensayos. La población de estudio fue cada elemento que compone los EPP. En la que una de sus principales conclusiones fue que gracias a la incorporación de un laboratorio se podrá evaluar pruebas en relación a accidentes eléctricos, cuyo propósito permite que los colaboradores puedan trabajar de manera eficiente y tener un desempeño lustre. En referencia, se puede intuir que si una empresa instituye este proyecto, tendrá mejoras en cuando a los índices por accidentes eléctricos, cuyo índice disminuirá garantizando la seguridad de sus trabajadores como familiares.

Alarcón y Poma (2021), Su proyecto tiene como propósito la incorporación de un algoritmo centrado en IA para la detección de EPP en entornos laborales

a través de sistemas de supervisión basados en Cámaras IP. La relevancia de esta iniciativa radica en mejorar de la seguridad y salubridad de los trabajadores, cumpliendo los protocolos de uso de EPP.

La metodología empleada se sustenta en el empleo de software de visión artificial y redes neuronales convolucionales, específicamente la red neuronal OpenPose, que permite extraer imágenes de las áreas convenientes, como la cabeza y las manos de los operarios. Estas ilustraciones se utilizan como datos que pasan por los sistemas neuronales convolucionales para identificar la existencia de EPP, como cascos y guantes.

Este proyecto se aplica en el contexto de la compañía “Distribución Eléctrica S.A.”, donde se implementa una aplicación computarizada que posibilita al consumidor observar las alertas producidas por el algoritmo y recepcionar notificaciones por correo electrónico. Los resultados indican una alta precisión en la detección de EPP, con una precisión de aproximadamente el 90% o superior.

En conclusión, el sistema propuesto se muestra eficiente y viable técnica y económicamente. Cumple satisfactoriamente con los objetivos propuestos en el proyecto, lo que lo convierte en una propuesta valiosa para mejorar la seguridad en el ámbito laboral y garantizar el uso adecuado de EPP. Este sistema puede ser aplicado en diversas empresas, aportando a la seguridad y confort de los trabajadores.

Gutiérrez (2022) En su proyecto tiene como objetivo mejorar la seguridad en la industria de la construcción al automatizar el reconocimiento y monitoreo del uso de EPP. Su metodología consta en la aplicación de tecnología de

aprendizaje profundo (deep learning), utilizando algoritmos como VGG-16, Resnet-18 e Inception-V3 para la detección de EPPs en imágenes bajo diversas condiciones visuales. El resultado del estudio demostró que el algoritmo Inception-V3 logra una precisión del 84% en condiciones regulares. Esto sugiere que el aprendizaje profundo puede contribuir significativamente al monitoreo y control del uso de EPP en entornos de construcción, siempre que se utilicen conjuntos de datos adecuados y se sigan las mejores prácticas en la implementación del modelo.

En conclusión, el uso de EPP es fundamental para la seguridad de los trabajadores en la construcción, pero su cumplimiento en el campo es desafiante debido a diversos factores. La tecnología de preparación puede ser una respuesta eficaz para automatizar la detección de estos equipos y prevenir comportamientos inadecuados. Además, este proyecto contribuye al conocimiento al proporcionar un resumen del estado actual de la visión computacional en la construcción y al crear una base de datos más amplia y variada de EPPs para futuras investigaciones en este campo.

2.1.3 Antecedentes locales

Vilca, (2019) En su plan *“Diseño e implementación de sistema de visión artificial con una cámara de escaneo lineal para la inspección de etiquetas en envases cilíndricos”*. Tiene como finalidad incorporar inteligencia artificial (visión) empleando una cámara de escaneo lineal para inspeccionar etiquetas de recipientes cilíndricos. Los cuales pueden controlar y sincronizar variables, tales como: perspectiva, tiempo de exposición de la cámara, iluminación y la transformación de señales e imágenes. En conclusión, se diseñó un sistema de visión artificial, la cual obedece los lineamientos necesarios de seguridad para

clasificar los recipientes según la fecha de vencimiento, escaneando el producto. Además de la sincronización y notificación son funciones que se adoptan de manera óptima.

Pacco (2022) El propósito de este proyecto fue desarrollar e implementar un sistema de videovigilancia (CCTV) con inteligencia artificial que pueda tomar decisiones automáticas, como activar alarmas y notificar eventos de robo en caso de detectar armas, con el fin de incrementar la efectividad de los sistemas de seguridad electrónicos y combatir la delincuencia en negocios, como joyerías. Se utilizó la Raspberry Pi 4 para aplicar visión artificial en la detección de armas y se estableció la notificación de eventos de robo a través de WhatsApp. Los resultados demostraron una precisión del 90.30% en la detección de armas, y el costo de implementación es asequible para pequeños y medianos negocios. Aunque la notificación por WhatsApp puede tener demoras debido a la interfaz gráfica, este sistema es eficaz para evitar robos al activar el sistema de alarma en aproximadamente 6 segundos.

En conclusión, se ha logrado desarrollar un sistema de videovigilancia inteligente eficiente y asequible para detectar armas. Este sistema tiene el potencial de mejorar significativamente la seguridad en diversos tipos de negocios, y su costo de implementación es razonable. Sin embargo, se debe tener en cuenta posibles demoras en las notificaciones por WhatsApp debido a la interfaz gráfica. En general, este proyecto representa un avance importante en la aplicación de la para la seguridad electrónica.

2.2 BASES TEÓRICAS

2.2.1 Equipos de protección personal

2.2.1.1 Clasificación

Para Abrego y Molinos, (s.f) determina que cualquier implemento de seguridad, que obvio el riesgo de sufrir un accidente, debe ser tratado y escogido por un personal profesional especializado, que respete los lineamientos de seguridad, es decir, deben estar prestos a las cumplir las condiciones del instituto Nacional de Normalización (INN) o organismos internacionales reconocidos.

2.2.1.2 Protección de cráneo

Según Abrego y Molinos, (s.f) son primordialmente los cascos, los cuales protegen de golpes, riesgos eléctricos, químicos y térmicos, los cuales se clasifican en:

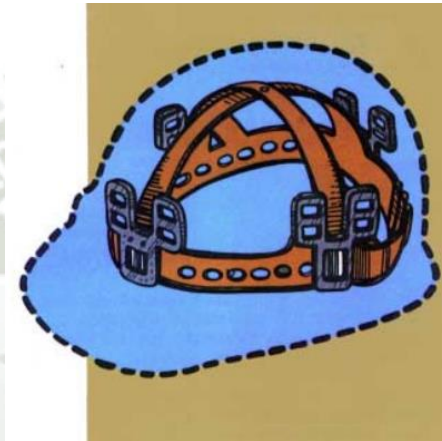
- A, aquellos cascos cuya función primordial es proteger contra salpicaduras de sustancias raras, impactos, lluvia, llamas e incluso capaz de soportar una tensión de 15.000 V con una corriente de fuga máxima de 8 mA y tolerar una tensión de hasta 20.000 V sin experimentar la falla del aislante.
- B, cumplen la misma función, pero toleran una tensión de 2.200 V con una corriente de fuga máxima de 3 mA.
- C, este tipo de implemento se clasifica en tercer lugar, es también capaz de soportar impactos, llamas, lluvia, salpicadura de sustancias, pero estos no se encuentran obligados a cumplir con las características dieléctricas.

- D, estos cascos solo cumplen con sola una única función, que es la de resguardar impactos, sin exigencias de otra índole.

Este tipo de cascos suelen ser metálicos, los cuales se pueden complementar con otros dispositivos e implementos de seguridad.

Figura 2

Partes constitutivas de los cascos



Nota. Abrego y Molinos, (s.f)

2.2.1.2.1 Protección de ojos y cara

Existen diversos elementos que protegen nuestros ojos sujetos a la forma y calidad de uso en protección, ya que depende de la condición de trabajo a realizar, y los peligros que se pueden sufrir sin este equipo de protección. Este elemento se clasifica en tres grupos:

- Contra proyección de partículas. Para trabajos manuales, sin protección.
- Contra humos, gases, vapores y líquidos.
- Contra radiaciones.

Figura 3

Protección de ojos



Nota. Abrego y Molinos, (s.f)

Por otro lado, los elementos de resguardo facial, que se clasifican en:

- Máscaras con lentes de protección (máscaras de soldador).
- Protectores faciales

Figura 4

Protección de rostro



Nota. Abrego y Molinos, (s.f)

En cuanto a los protectores faciales, estos elementos favorecen y nos resguardan de la propagación y el contacto contra partículas y otros cuerpos extraños, que pueden dañar el rostro, esencialmente, los ojos. Este elemento se presenta en diferentes materiales de fabricación, que puede ser plástico, cristal templado, o puede ser metálica.

2.2.1.2.2 Protección de manos y brazos

Según Abrego y Molinos, (s.f) las manos son herramientas fundamentales e imprescindibles para realizar un trabajo, por ello también se encuentran expuestas a sufrir lesiones o peligros concurrentes como cortes, quemaduras, golpes, etc., debido a su constante uso activo en los diversos procesos de producción en especial, en los puntos de operación con maquinaria. Ciertos análisis estadísticos indican que, de todos los incidentes, accidentes que se dan en el trabajo, un 30% de estos afectan las manos y brazos. Por ello es necesario y fundamental proteger los brazos y manos de posibles riesgos, tales como materiales o sustancias calientes, cortantes y disolventes, corrosivos, abrasivos, electricidad, chispas de soldaduras, etc., usando guantes, que dependen del trabajo a realizar. Es más, existen tipos de guantes:

- Guantes de cuero curtido al cromo: Se emplean para evitar raspones o lesiones por fricción, siendo fundamentales para todos los trabajadores.
- Guantes de goma pura: Generalmente este tipo de guantes se usa en trabajos sometidos a corriente eléctrica.
- Guantes de material sintético: Básicamente los más conocidos y frecuentemente empleados en actividades donde se elaboran productos químicos, como: ácidos, solventes y aceites, pues estos son caucho, PVC y neopreno.
- Guantes de asbesto: Estos guantes son de un material que soportan altas temperaturas. Suelen ser usados por soldadores, fundidores, fogoneros, etc. trabajadores que están expuestos a

manipular metales u otros elementos a temperaturas elevadas (calientes)

2.2.1.3 Protección de pies y piernas

Según Abrego y Molinos, (s.f) las piernas y pies son partes del cuerpo que se deben de cuidar frente lesiones que pueden causar la caída de objetos puntiagudos, pesados, objetos que ruedan o vuelcan, cortes, efectos corrosivos de productos químicos, etc. por ello es fundamental el uso de zapatos, no cualquier zapato, sino unos especiales, llamados zapatos de seguridad, que están hechos de diversos materiales y formas, con el único objetivo de cuidar los pies. El calzado está constituido por:

- Puntera o casquillo de acero, la sección en la parte frontal del calzado resguarda los dedos de los pies de cualquier impacto o compresión.
- Suela de goma o PVC, esta suele ser de un grosor estandarizado, de tal manera que no pueda consumirse muy rápido en su uso, además de cuidar la base del pie (planta), ayuda a evitar caídas o resbalones (tiene un buen agarre), además de posibles cortes.
- Cubierta, es de cuero grueso con una gran resistencia al impacto, cortes, rasgadura, fluidos ácidos, aceites, solventes, etc. Incluso, entre la plantilla y la suela se encuentra una separación aislante de corcho.
- Los equipos de protección personal para los pies son:
 - Calzado de protección
 - Calzado de seguridad.
 - Calzado de trabajo.

- Calzado y cubre calzado de protección contra el calor.
- Calzado de protección contra las motosierras.
- Calzado y cubre calzado de protección contra el frío.
- Calzado frente a la electricidad.
- Protectores amovibles del empeine.
- Polainas.
- Suelas amovibles (antiperforación o antitranspiración, antitérmicas).
- Rodilleras.

2.2.1.3.1 Ropa protectora

Según Abrego y Molinos, (s.f) la ropa es un implemento fundamental para el trabajador, ya que depende de ella saber cuan expuesto se encuentra la piel, ya que el resto del cuerpo esta protegido solo por este elemento. Por ejemplo, se puede sufrir la exposición de la piel al sol, al polvo, grasa, aceite e incluso sustancias corrosivas o cáusticas. La ropa protectora también se encuentra clasificada, esta depende del tipo de material:

- Tejido: Las prendas de tela se emplea en situaciones donde se necesita una leve protección, especialmente contra el material particulado (polvo), contra el líquido de la pintura al pintar con la pistola y en trabajos específicos con abrasivos. El material más comúnmente usado es el algodón de tejido ajustado, y el diseño preferido es el overol con puños ceñidos en las muñecas y tobillos.
- Cuero: Se emplea comúnmente para prendas que tienen la función de proteger una cierta parte del cuerpo específicamente, como

ropa utilizada en trabajos de manipulación manual, mandiles de soldador, zapatos, etc. El cuero puede tratarse como un material a prueba de grasa, un material no filtrante.

- Caucho: Este material es utilizado exclusivamente como parte de la confección de una vestimenta. Los mandiles de caucho normalmente se elaboran a partir de láminas de goma en una cara o simultáneamente las dos caras.

- Plásticos: Dicho material constituye trajes que proporcionan resguardo contra los elementos tóxicos, corrosivos o causticas, e incluso inclemencias del tiempo (humedad, sol, etc.). Por ejemplo, los delantales o mandiles, suelen hacerse de este material plástico, es decir, se utilizan aminoras de PVC o de fibra sintética o artificioso.

2.2.2 Visión artificial

Se trata de una rama de la IA que, gracias al manejo de diferentes mecanismos; se puede acceder a la obtención de datos, su procesamiento y evaluación de diferentes clases de información o data; dicha información se obtiene por imágenes digitales. Básicamente se considera una especialidad y forma parte de la ingeniería, informática e incluso óptica. En otras palabras, la visión artificial es la competencia de una IA capaz de visualizar y apreciar el exterior y todo lo que le rodea de manera real (recibe y capta el entorno del medio), en ese sentido, esta IA obtiene información y la procesa, para obtener características bidimensionales o tridimensionales de imágenes digitales. (Martinez, 2018)

2.2.2.1 Visión

Según Sucar, (s.f.) La "Visión" es la ventana al mundo de la mayoría de organismos. Su tarea primordial es el de identificar y ubicar materiales u objetos en el medio ambiente a través del procesamiento de las fotografías, e ilustraciones digitales. La visión computacional se encarga de analizar estos procesos, para poder comprender, asimilar el comportamiento y fabricar máquinas con aptitudes similares.

Para Marr, (1982) "visión es un proceso que proporciona, a través de las imágenes digitales, una caracterización detallada del mundo externo siendo útil para el observador y que tiene data significativa".

La conceptualización que tiene más influencia y relevancia sobre la idea actual de una visión computacional es la explicación de (Marr, 1982).

Esta explicación considera 3 aspectos primordiales:

1. La visión como un proceso computacional.
2. La caracterización detallada a obtener depende totalmente del observador.
3. La necesidad de descartar la información no válida o no útil (educación de información)

2.2.2.2 Etapas de la Visión Artificial

Un sistema de visión artificial implica varias técnicas, que cumplan con la función del procesamiento gráfico de imágenes (obtención, codificación, transformación de imágenes) o incluso el reconocimiento de formas en la vida real (teoría estadística de decisiones y diversas metodologías, como enfoques sintácticos y neuronales, que se utilizan para la clasificación de patrones.).

Figura 5

Etapas fundamentales para el sistema de visión artificial



Nota. Pichucho et. al, (2018)

En este tipo de sistemas, también incorporan técnicas y /o mecanismos que permitan el modelado geométrico y procesos de conocimiento. En referencia, es verdad que este tipo de especializada artificial posee características específicas y necesarias para lograr su objetivo y función, también es cierto que la mayoría de máquinas que se fabrican bajo IA tienen procesos comunes en su realización (Jiménez, 2015)

2.2.2.2.1 Adquisición

Etapa donde se obtienen las imágenes. La finalidad es resaltar y acentuar a través de técnicas fotográficas, las cualidades visuales de cualquier objeto. Es tal vez la etapa primordial de todo el proceso, y gracias a la calidad de las imágenes podemos construir y traducir información, una buena adquisición de imágenes permite poder continuar con la siguiente etapa, es decir, poder procesar. (Jiménez, 2015)

Figura 6

Imagen Original



Nota. Jiménez, (2015)

2.2.2.2.2 Pre - procesamiento

La finalidad de esta etapa es mejorar la calidad de la imagen y así poder obtener información verdadera, y detallada. Para ello, se usan mecanismos que permitan acentuar diferentes aspectos, mediante la eliminación de ruido, reafirmar el contraste, perfeccionar la intensidad y extraer el borde de la imagen.

Figura 7

Imagen procesada de un objeto



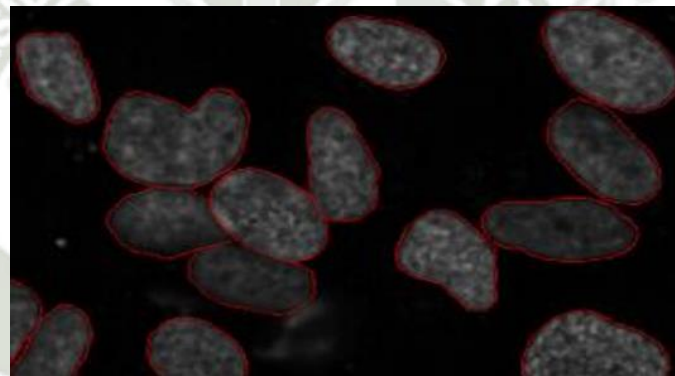
Nota. Jiménez, (2015)

2.2.2.2.3 Segmentación

Se fracciona la imagen en partes con el fin de extraer la información, en otras palabras, dividir la imagen en partes para realzar su importancia. Estas partes son homogéneas según sus características, ya sea en el color, textura e intensidad en base a detalles de discontinuidad y similitud. La adquisición de algunas partes y el descarte de otras se definirá en base al objetivo único del uso, identificando los objetos de interés.

Figura 8

Segmentación nuclear de células



Nota. Jiménez, (2015)

2.2.2.2.4 Descripción

En esta etapa se conceptualiza toda la información que se puede sustraer del objeto a analizar, las características y/o cualidades detalladas que marcan la diferencia, atributos invariables, es decir, aquellos que no cambian ante rotaciones o escalas, como por ejemplo el perímetro del contorno, la textura, etc.

2.2.2.2.5. Redes Neuronales artificiales

Un tipo de diseño de red neuronal artificial más avanzado que el perceptrón multicapa es la red neuronal convolucional CNN. Este tipo de red se construye como una variación del perceptrón multicapa. Una red neuronal

artificial (RNA) normal se distingue de ésta por el hecho de que emplea capas de convolución y submuestreo para gestionar con mayor eficacia las fotos y vídeos que se le introducen.

Dado que este tipo concreto de arquitectura de red se estableció especialmente con el fin de tratar datos bidimensionales, se utiliza a menudo en el uso de espectrogramas para la identificación de señales de voz e imágenes. Esto se debe al hecho de que se construyó expresamente para el propósito descrito anteriormente. Además, si se alteran, son capaces de funcionar en más de dos dimensiones con los ajustes adecuados. Cuando Kunihiko Fukushima empezó a trabajar en una red neuronal que modeliza el funcionamiento del córtex visual en 1982, ideó la iteración inicial de este sistema. Este sistema sigue utilizándose hoy en día (ESCOBAR, 2023).

Las redes neuronales convolucionales, más conocidas como CNN, son un tipo de red neuronal artificial que surgió como consecuencia del crecimiento del perceptrón multicapa a lo largo de su evolución. Del mismo modo que otros tipos de redes neuronales están formadas por una serie de neuronas artificiales, las CNN son un tipo de red neuronal. Según Freeman (1991), todas y cada una de las neuronas artificiales son responsables de una función bastante fácil de entender. Además de ser responsable de modificar los pesos de las neuronas, también es responsable de aceptar impulsos de otras neuronas, así como de entradas externas. También es responsable de calcular una señal de salida que luego se utiliza para alimentar a otras neuronas. El trabajo que realiza cada neurona y/o perceptrón está estructurado de una determinada manera (Alarcón & Poma, 2021).

Para el autor Martin (2023) El funcionamiento de las redes neuronales convolucionales, también conocidas como CNN, se basa en el concepto de convolución, que es una simulación de las operaciones que realiza el cerebro humano para identificar características visuales. En lugar de tratar cada píxel por separado, las redes neuronales convolucionales (CNN) aplican filtros convolucionales, a menudo denominados kernels, a la imagen de entrada. Estos núcleos son matrices de valores que se desplazan por la imagen para reconocer determinados patrones en áreas de la imagen que son diferentes entre sí. En estos diseños se incluye una gran variedad de formas, texturas y bordes. El uso de distintos filtros da lugar a la formación de estos mapas de características, que resaltan determinados aspectos de la imagen. Estos mapas se integran en otras capas de la red para llevar a cabo tareas como la clasificación, la identificación de elementos o la segmentación. El objetivo de esta integración es completar estas tareas. En una arquitectura convolucional, los siguientes tipos de capas son los que se encuentran la mayoría de las veces:

Las capas que se construyen a partir de bloques de kernels se denominan capas convolucionales. El objetivo de estas capas es extraer la información relevante de los datos, como ya se ha dicho. Las primeras capas se dedican a la extracción de características de bajo nivel, mientras que las siguientes se centran en las características de alto nivel. Las características de bajo nivel se extraen de los datos. Se realizan iteraciones para llevar a cabo la operación de convolución entre la matriz de entrada y cada núcleo individual. Esta operación se realiza para obtener el resultado deseado (Martin, 2023).

Una serie de iteraciones de activación: Una vez finalizado el procedimiento de convolución, es habitual aplicar una función matemática que

incorpora una transformación no lineal para conseguir la convergencia del modelo. Esto se hace para garantizar que el modelo sea similar al original (Martin, 2023).

Capas de agrupamiento: Estas capas disminuyen la dimensionalidad espacial de las características, lo que a su vez reduce el número de parámetros de la red, que a su vez hace que la red sea más eficiente. Las capas de agrupamiento reducen aún más el número de parámetros de la red. Es práctica habitual elegir el valor máximo (max pooling) o el valor medio (average pooling) de una colección de datos que se sitúan dentro de una región al hacer el pooling. Esto se debe a que ambos valores se consideran los más precisos. De este modo, es posible mantener las características más cruciales y, al mismo tiempo, reducir la sensibilidad a los pequeños cambios (Martin, 2023).

La normalización y la regularización se refieren a las capas siguientes: Además, el uso de estas capas se traduce en una mejora tanto de la generalización como del rendimiento de la red. Como resultado de la capa de normalización por lotes, las activaciones de cada capa se normalizan, lo que no sólo hace que el proceso de entrenamiento sea más rápido, sino que también reduce el riesgo de que se produzca un sobreajuste. Un porcentaje específico de neuronas se desactiva aleatoriamente durante el proceso de entrenamiento para evitar una dependencia excesiva entre ellas. Esto se consigue mediante el uso de una técnica de regularización que se conoce como capa de desactivación (Martin, 2023).

Capas adicionales que están completamente conectadas: Después de estos niveles, es posible añadir una o más capas adicionales completamente conectadas. En una red neuronal tradicional, cada neurona está conectada a

todas las neuronas de la capa anterior. Las capas que componen esta red son análogas a las que componen una red neuronal convencional.

La capa de aplanamiento, que a veces se denomina capa de aplanamiento, crea un vector unidimensional. Esta capa se encarga de convertir las matrices de características. Es esencial que esto se haga para garantizar que la información pueda transmitirse a los niveles completamente conectados que se sitúan en las capas futuras de la arquitectura de red. Para realizar la transición de las características obtenidas en las rondas anteriores a los juicios finales de clasificación o regresión, la capa Flatten es un paso crítico que debe darse (Martin, 2023).

2.2.2.2.5 Reconocimiento

En esta etapa se clasifica e identifica los objetos de la imagen, y para ello, los mecanismos más utilizados son algoritmos genéticos, métodos estadísticos y redes neuronales.

2.2.2.2.6 Interpretación

En esta etapa se busca dotar de significado a los objetos reconocidos, es decir, proporcionar información apto y verdadero de las imágenes a estudiar.

2.2.3 Cámaras

La cámara es un equipo óptico que permite captar la imagen que se proyecta en el sensor para ser transmitido a una red electrónica. Las cámaras empleadas en visión artificial poseen unas cualidades específicas que permiten capturar momentos accediendo a un control del disparo en la orientación que se requiera. Las cámaras para visión artificial marcan una gran diferencia de las

cámaras comunes que conocemos, ya que disponen de un dominio cabal de: tiempos, velocidad de obturación, señales, sensibilidad, etc. (Vilca, 2019)

2.2.3.1 Importancia

Trabajar con la visión artificial requiere de un aparato que perciba la luz, con el objetivo primordial de guardar y resguardar las imágenes en formato digital. Es decir, es primordial adquirir e implementar una cámara de vídeo, cámara web o una capturadora analógica. Las cámaras web se pueden ensamblar directamente a las computadoras mediante el acceso de dispositivos USB, FireWire o Thunderbolt; y consiguiendo capturar sus fotogramas en tiempo real. (Jiménez, 2015)

2.2.3.2 Cámaras matriciales

Según Vilca, (2019) las cámaras matriciales están constituidos por un sensor CCD o CMOS matricial, cuyo objetivo es analizar las imágenes en 2D (bidimensional). Los dispositivos CCD consisten en una disposición en forma de matriz de diodos fotosensibles. Estos sensores presentan algunas características, tales como:

- Factor de relleno: Porcentaje del área del pixel sensible a la luz, a causa del apartamiento entre los registros que no llega al 100%.
- Tipo de transferencia: Existen varios tipos, según la forma de transferencia de la data.
 - Transferencia de cuadro: Emplean una región específica para la captura de luz, aparta del área activa. Esto permite lograr un mayor factor de llenado, pero con la desventaja de reducir la velocidad de obstrucción.

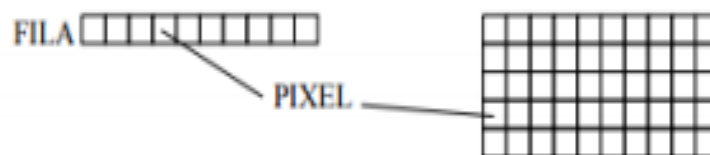
- Cuadro entero: Emplean un padrón para la manifestación de fotones, incorporación de la carga y traslado de la misma; consiguiendo un factor de relleno de 100%.
- Transferencia Inter-Línea (ITL): Son los más usuales, emplean un padrón de desplazamiento localizados entre las líneas de pixel para acopiar y traspasar la información, lo que concede una alta velocidad de obstrucción.

2.2.3.3 Cámaras lineales

Están constituidos por un sensor CCD lineal. Básicamente, las cámaras lineales configuran todas las imágenes línea a línea efectuando una eliminación de manera perpendicular al sensor lineal.

Figura 9

Construcción lineal de imagen

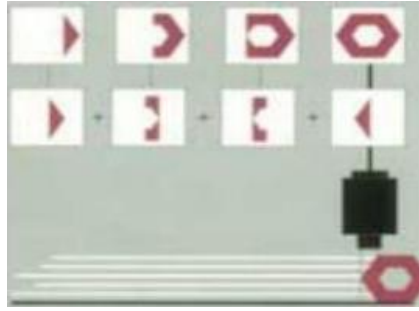


Nota. Vilca, (2019)

Producir imágenes de alta calidad a partir de tomas lineales individuales demanda una exactitud significativa. Por esta razón, tanto la alineación como el sincronismo son aspectos esenciales y delicados para obtener una imagen que resalte los atributos del objeto.

Figura 10

Obtención de la imagen mediante barridos sucesivos



Nota. Vilca, (2019)

Características técnicas de las cámaras lineales según Vilca, (2019):

- Número de componentes del sensor: El número de píxeles determina el tamaño de la óptica.
- Velocidad: Se termina como la lectura de píxeles por unidad de tiempo. Estas pueden alcanzar velocidades de hasta 200 MHz, superiores a las de las cámaras matriciales.
- Cámaras lineales a color: Están constituidas por tres sensores lineales para cada color (rojo, azul y verde). Según su progreso o evolución, estas pueden clasificarse en 2 categorías:
 - Trisensor: En este diseño, los sensores están dispuestos en proximidad estrecha para lograr una sensibilidad óptima, empleándose únicamente en superficies planas.
 - Prisma: En esta configuración, los sensores se encuentran en las tres caras de un prisma, requiriendo una iluminación más intensa sobre el objeto.

2.2.3.4 Cámaras inteligentes

La creación de nuevos componentes de los dispositivos de cámaras, son cada vez más pequeños, lo cual han desarrollado y proliferado a la fabricación de nuevas cámaras inteligentes. Estas cámaras están constituidas por

implementos similares a los de las cámaras comunes como la electrónica asociada a la captura y el sensor; sin embargo, también tienen incorporado una memoria, un procesador y un sistema de comunicación con el entorno (Ethernet y Puerto Serie. I/O), dando así una cámara con un mejor sistema, completo y eficaz (Jiménez, 2015)

2.2.3.5 Cámaras IP

Según Obregón, (2017) Las cámaras IP, conocidas como cámaras de red o digitales, ofrecen capacidades de resolución de imagen, control de eventos, conectividad y video inteligente que no están disponibles en las cámaras analógicas. Esto permite identificar:

- El desplazamiento de una persona u objeto en el video.
- Las Alteraciones o sabotajes a la cámara desde sistemas externos que se encuentren conectados al puerto de entrada/salida (como interruptores de puertas, activación de alarmas o sensores de movimiento PIR).
- Aviso de eventos utilizando SMTP, FTP, HTTP (SMS, MMS, email, etc.) Power over Ethernet (PoE) señal y alimentación por el mismo cable Ethernet.
- Afinidad para la entrada de dispositivos, desde móviles inteligentes.
- Audio y/o sonido bidireccional sobre el mismo sistema Ethernet (SIP, IFX).
- Iluminadores infra rojos para la adquisición de imágenes tomadas durante el día, y tomadas durante la noche.

- Comandos PTZ remitidos por el cable de red de manera automática o manual.
- Recorte y detalle (zoom digital) de la función en la imagen que está captando.
- Grabación de imágenes.

2.2.3.6 Cámara Web o Webcam

Es un aparato que posee entrada USB, el cual se conecta con la computadora, cuya finalidad es tomar fotos digitales o grabar videos, pero con una resolución baja, el cual ha sido reemplazado por la videocámara digital. El video que captura y procesa, lo codifica de manera específica para enviarlo por Internet de manera instantánea hacia otra computadora para que el otro usuario pueda visualizar el video en el mismo instante; son utilizadas para comunicarse de manera personal y virtual haciendo uso del internet, también se pueden utilizar para funciones de vigilancia. (Jiménez, 2015)

Figura 11

Cámara web o webcams



Nota. Acuña, (2018)

2.2.4 Detección de forma y color

2.2.4.1 Detección basada en color

El color resulta crucial para la identificación detallada de imágenes. Para emplear este método, se inicia con la conversión del espacio de color mediante el uso de cámaras RGB a color. Algunos expertos en este campo expresan preferencia por utilizar el espacio de color RGB o una variante ajustada, mientras que otros optan por la conversión de espacios de color para obtener resultados más óptimos (Vargas, 2015).

2.2.4.2 Segmentación por Umbralización de Color

Esta técnica es la más sencilla de realizar para efectuar la segmentación de imágenes de color. Para ello, se requiere emplear un vaso umbral que permitirá la clasificación de los píxeles de una imagen, facilitando los píxeles del área que nos importa convirtiéndose el resto en el fondo de la imagen no necesario. El color de referencia de cada señal es usado para definir el umbral de cada Píxel.

2.2.4.2.1 Agregación de píxeles Dinámico

Este método incluye un umbral dinámico, el cual facilita la segmentación midiendo el proceso de incorporación de píxeles utilizando el espacio del color HSV. La principal y gran ventaja del umbral dinámico es disminuir la inestabilidad de tono y la modificación de brillo externo.

2.2.4.2.2 Transformación HSV a HSI

Estos modelos de color separan la información sobre la intensidad global del color en tonalidad y saturación, lo que los vuelve menos susceptibles a cambios provocados por la iluminación. La modificación de espacio de color RGB a HSI se consigue cuando se separa la información cromática e información de

intensidad de color, el sistema HSI percibe los colores casi idénticos a la percepción del ojo humano.

2.2.4.2.3 Crecimiento de Región

Esta técnica emplea una semilla en un área como partida y se extiende como grupos de píxeles según los parecidos de color. Esta técnica se puede amplificar en el espacio de color HSI, además puede provocar inconvenientes cuando las condiciones de término no son complacidas, ya que se usa una semilla de arranque y concluye cuando se cumplen ciertos parámetros. (Vargas, 2015)

2.2.4.3 Detección basada en forma

Se emplean imágenes en escala de grises para eludir inconvenientes relacionados a colores, para detectar las señales se utilizan los bordes exteriores. Posteriormente, se hace una síntesis de ciertas técnicas de detección de señales más raras en la forma.

2.2.4.3.1 La Transformada de Hough

Esta técnica se emplea para identificar características habituales como líneas y círculos, pero su complejidad computacional demanda una gran cantidad de memoria, lo que la hace menos adecuada para aplicaciones en tiempo real.

2.2.4.3.2 Detección por Similitud

Este método se basa en la perspectiva óptica donde se requiere buscar un factor de similitud (idéntico) entre un área segmentada y una agrupación de imágenes binarias. Donde representan las formas de cada señal. El método asume que la imagen ejemplar y la segmentada tienen las mismas dimensiones.

2.2.4.3.3 Detección por características espaciales

La búsqueda de formas geométricas se fundamenta en las propiedades espaciales de las señales presentes en una imagen capturada. Una vez identificada esta forma, se genera una lista para su posterior clasificación.

2.2.5 Reconocimiento de patrones

2.2.5.1 Detección de objetos

Al proporcionarnos una imagen para estudiarla, primero se debe aplicar un algoritmo de identificación de todas las posibles sub - ventanas en la imagen. Dichos algoritmos están predispuestos a ser algo lentos, además de tener una tendencia a presentar desaciertos. Sin embargo, es más eficiente desarrollar detectores con un propósito definido, cuya labor es descubrir diligentemente probables áreas donde pueden presentarse objetos exclusivos. (Estela, 2016)

2.2.5.2 Características vectoriales

Se trata de describir las cualidades y características fundamentales como la textura, el color, la forma, el desplazamiento, etc. El argumento de una imagen se encuentra codificado digitalmente especificado en cada unidad mínima de la imagen; dicha información obtenida pertenece a cada pixel que compone dicha imagen. De tal manera los pixeles expresan la conexión entre las características propias de una imagen y el argumento abstracto de sus valores. Es así que, cualquier técnica de control de imágenes que se rige en su argumento debe almacenar un lazo de relación sobre el valor de los mismos. (Estela, 2016).

2.2.6 Algoritmo LBPH

El algoritmo LBPH (Local Binary Pattern Histogram) es una técnica de reconocimiento de patrones utilizada en el ámbito de la visión por computadora

y el análisis de imágenes. A diferencia de algunos algoritmos de reconocimiento facial más recientes, no hay un solo autor o investigación pionera que se pueda atribuir al desarrollo exclusivo del algoritmo LBPH. En cambio, este método es el resultado de una evolución de la visión por computadora, con contribuciones de varios investigadores a lo largo del tiempo.

Aunque no hay un único autor que haya creado el LBPH, hay estudios y trabajos que han desempeñado un papel fundamental en su desarrollo y aplicaciones posteriores. Uno de estos estudios se titula "Face recognition with local binary patterns" y fue realizado por Timo Ahonen, Abdenour Hadid y Matti Pietikäinen. Este estudio se centra en la utilización del LBPH para el reconocimiento facial, destacando sus ventajas y aplicaciones en este contexto. Sin embargo, no se puede considerar como la fuente única o la única investigación relevante en relación con el LBPH.

El concepto detrás del algoritmo LBPH implica la descripción de texturas en una imagen mediante patrones binarios locales. Funciona examinando cada píxel de la imagen y comparándolo con sus vecinos en un patrón determinado. A partir de estas comparaciones, se crean histogramas que representan las texturas en la imagen. El LBPH es ampliamente utilizado en aplicaciones de reconocimiento facial, detección de objetos y más, y ha demostrado ser efectivo en una variedad de contextos.

2.2.7 Ordenador de programación

La programación de computadoras no debiera proyectar de manera dividida del estudio de los requisitos e implementos necesarios del sistema de información que se quiere mecanizar. La supuesta evidencia de dicha afirmación

no es la misma si valoramos que la programación está logrando conseguir un carácter masivo de uso al ser empleada por profesionales y estudiante de diversas edades, además existe una relación y una diferencia entre el análisis, que se basa en la descripción del problema, y la programación, que se refiere en sí a la codificación. Hoy en día, las diversas actividades de estudio y modelamiento en tecnología de la información continúan avanzando (Verónica y Ángelo, 2017)

2.2.8 Microcontroladores Raspberry Pi OS

Este dispositivo funciona bien como multiprocesador. Eso tiene una tarjeta gráfica, RAM, una memoria volátil, dispositivo interfaces y otras interfaces de equipos inalámbricos externos. Esta raspberry Pi consume muy menos energía, pero es todavía barato y efectivo. Se necesita de un teclado con el que se emite comandos, un monitor y el power supply de serie ORDENADOR PERSONAL. Aquí, Raspberry Pi usó la tarjeta SD como disco duro. Raspberry Pi para enlazarse a través de una LAN / Ethernet o mediante un Módem USB o vía inalámbrica. Se supone que Raspberry Pi soporte para diversas aplicaciones domésticas y comerciales. Raspberry Pi se ejecuta en una red operativo centrada en Linux y que operado por la red operativa Raspbian. Python es una programación lenguaje utilizado para implementar la Raspberry-Pi. Tiene la capacidad de comunicarse con otros dispositivos externos. (Seena y Sudershan, 2019)

Raspberry Pi OS, denominado como Raspbian, es una variante del sistema operativo GNU/Linux basada en Debian y especialmente diseñada para la SBC (Single Board Computer) Raspberry Pi, dirigida a la educación en informática. Fue lanzado en junio de 2012 y a partir de 2015, la Raspberry Pi

Foundation promovió oficialmente como el sistema operativo principal para la serie de placas SBC de Raspberry Pi. (Raspbian, 2021).

Existen numerosas variantes de Raspbian, siendo la más actual la Raspbian Buster

Figura 12

Interfaz de Raspberry Pi OS



Nota. FrontPage - Raspbian. www.raspbian.org.

Raspbian fue desarrollado por Peter Green y Mike Thompson, como un proyecto separado. La primera versión se lanzó en junio del 2012. Este Sistema operativo se encuentra aún en desarrollo. Según el sitio web Raspbian (2021), este está enormemente optimizado para Raspberry Pi.

Técnicamente, este sistema operativo es un port de Debian armhf no oficial para el procesador Raspberry Pi (CPU), con soporte óptimo para cálculos de punto flotante por hardware, lo que permite una amplia variedad de aprovisionamiento y más rendimiento en diferentes circunstancias. Según Raspbian (2021), el port es necesario porque no hay una versión de Debian armhf para CPU ARMv6 que contengan Raspberry Pi.3.

El menú "raspi config" también es destacable, lo que le permite una configuración del sistema operativo sin tener que cambiar manualmente el archivo de configuración. Entre sus funcionalidades, nos posibilita extender la partición root para que necesite la tarjeta de la memoria, aplicar overlocking, configurar el teclado, etc. (ELINUX, 2020)

En el 2012, con la versión 2012-12-16-wheezy-raspbian, fue lanzado a la tienda de aplicaciones "Pi Store", que en ese momento del lanzamiento incluían aplicaciones como LibreOffice o Asterisk para juegos como OpenTTD o Freeciv. En esta plataforma, el contenido gratuito o de pago, como binarios, código Python, imágenes, audios o videos, pueden ser colocados a disposición y requerimientos de los usuarios de Raspbian. También se pensó en incluir documentación sobre Raspberry Pi, como la revista MagPi y las instrucciones del proyecto. En la actualidad, la tienda ya no está incluida con el sistema, pero puede ser instalado ejecutando el comando "sudo aptget update y sudo aptget install pystore"

Como distribución de GNU/Linux, ofrece numerosas posibilidades. Todo el software de código abierto puede ser compilado nuevamente en la Raspberry Pi para la arquitectura armhf, permitiendo su utilización en el dispositivo incluso cuando el creador no provea una versión compilada específicamente para esa arquitectura. Además, esta versión cuenta con repositorios que ofrecen a los usuarios la descarga de una amplia gama de aplicaciones, similar a una distribución de escritorio de GNU/Linux. Esto convierte a la Raspberry Pi en un dispositivo que, además de que te sirve como una placa microcontroladora clásica, tiene muchas funciones de una computadora personal. Esto podría convertirlo en una alternativa a las computadoras personales, en especial para

aquellos con escasos recursos, para expandir la informática en países menos desarrollados, o para aplicaciones más o menos costosas.

2.2.8.1 Antecedentes

El Dr. Eben Upton en el 2006 empezó a definir e introducir la programación y la electrónica. Estos autores notaron que los estudiantes no tenían habilidades computacionales, por ello se encontraban alarmados.

El Dr. Upton dispuso fabricar un ordenador rentable y económico, ya que exponía que los padres no tenían la posibilidad de adquirir computadoras para sus hijos por el alto costo, además de que este sea una razón para que estos últimos no se encuentren interesados en la programación. Esta concepción condujo al desarrollo de la placa Raspberry Pi, pues la computadora brindaría una extraordinaria conveniencia para los niños, ya que esto permitía instruirse, programando y cultivando sus contenidos personales.

El Dr. Upton junto a otros asociados fundaron Raspberry Pi, con la finalidad de impulsar la alfabetización y el mundo informático; en la actualidad los niños y jóvenes utilizan estas computadoras como su plataforma primaria.

Figura 13

Logo Raspberry Pi Foundation



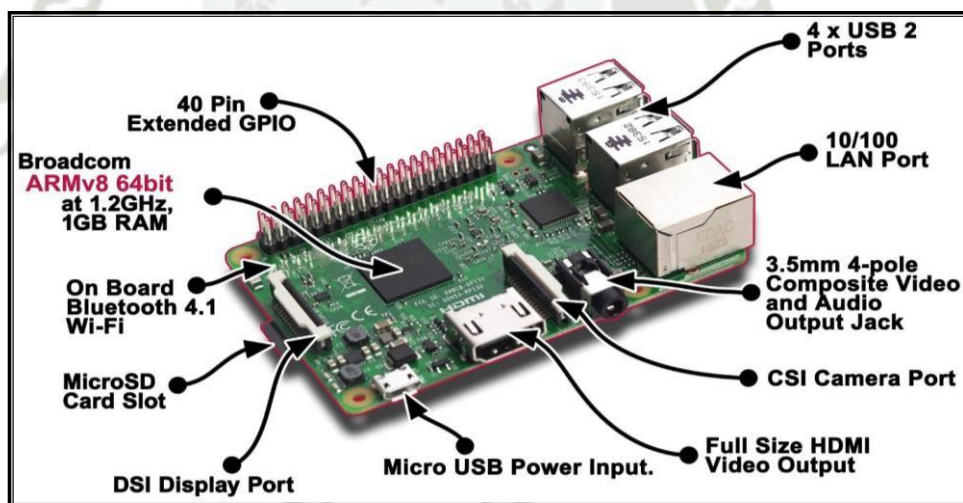
Nota. Raspbian (2021)

Incorporar un chip SoC en el Raspberry Pi permitió reducir los costos de diseño. SoC se entiende como System on a Chip, un dispositivo tecnológico de silicio que incorpora físicamente otros componentes del sistema; es decir acopia la memoria, el procesador gráfico y el microprocesador; en otras palabras, reduce las huellas de interconexión PCB y el circuito impreso.

Por otro lado, la fundación se asoció con Broadcom con el fin de emplear sus diseños en los microprocesadores y procesadores SoC. El SoC, además también en ciertos componentes fundamentales y conexiones que describen en la Figura 14.

Figura 14

Componentes De Una Raspberry Pi



Nota. Amazon (2021)

Aunque, quizá, no es esencial comprender el funcionamiento del microprocesador Broadcom para usar el Raspberry Pi, es necesario evaluar el comportamiento para diferenciar la calidad y potencia, además de entender porque este es más pausado que una PC y por qué se usa la cantidad mínima de energía para promover el trabajo de la placa.

2.2.8.2 Definición de Raspberry Pi

La Raspberry Pi es un ordenador económico que tiene un tamaño similar al de una tarjeta de crédito. Fabricado y lanzado por la Fundación Raspberry Pi, su objetivo principal es fomentar y mejorar la educación sobre el conocimiento de la informática.

Esta placa tiene dimensiones de aprox. 8.5 cm de largo por 5.3 cm de ancho. Dentro de su estructura, contiene un chip Broadcom BCM2835 que incorpora un procesador ARM11, proporcionando varias frecuencias de funcionamiento y la capacidad de realizar overclocking. Además, cuenta con un procesador gráfico VideoCore IV y, dependiendo del modelo, varía la cantidad de memoria RAM, que oscila entre 256 MB y 1 GB.

En el caso de video presenta una salida para audio mediante un minijack y una salida HDMI y un puerto Ethernet de 10/100 o conexión Wifi para el internet.

2.2.8.3 Modelos de Raspberry Pi

Raspberry Pi dispone de diversos diseños, que se distinguen por su capacidad, calidad y potencia; a continuación, en las siguientes figuras de muestra los diversos diseños o modelos de Raspberry Pi que proporcione al mundo.

Figura 15


Modelos De Raspberry Pi (A).

				
	Model A	Model A+	Model B	Model B+
SoC	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835
CPU	700MHz ARM1176JZF-S	700MHz ARM1176JZF-S	700MHz ARM1176JZF-S	700MHz ARM1176JZF-S
GPU	VideoCore IV	VideoCore IV	VideoCore IV	VideoCore IV
RAM	256Mb	256Mb	512Mb	512Mb
USB	1	1	2	4
Vídeo	RCA, HDMI	Jack, HDMI	RCA, HDMI	Jack, HDMI
Audio	Jack, HDMI	Jack, HDMI	Jack, HDMI	Jack, HDMI
Boot	SD	MicroSD	SD	MicroSD
Red	-	-	Ethernet 10/100	Ethernet 10/100
Consumo	300mA / 1,5w / 5v	400mA / 2w / 5v	700mA / 3,5w / 5v	500mA / 2,5w / 5v
Alimentación	MicroUSB / GPIO	MicroUSB / GPIO	MicroUSB / GPIO	MicroUSB / GPIO
Tamaño	85,6 x 53,98 mm	65 x 56 mm	85,6 x 53,98 mm	85 x 56 mm

Nota. P.E. (2015)

Figura 16

Modelos De Raspberry Pi (B)

				
	2 Model B	Zero	3 Model B	Zero W
SoC	Broadcom BCM2836	Broadcom BCM2835	Broadcom BCM2837	Broadcom BCM2835
CPU	900MHz Quad-core ARM Cortex-A7	1GHz ARM1176JZF-S	1.2GHz QUAD ARM Cortex-A53	1GHz ARM1176JZF-S
GPU	VideoCore IV	VideoCore IV	VideoCore IV	VideoCore IV
RAM	1Gb	512Mb	1Gb	512Mb
USB	4	1 Micro	4	1 Micro
Vídeo	Jack, HDMI	Mini HDMI	Jack, HDMI	Mini HDMI
Audio	Jack, HDMI	Mini HDMI	Jack, HDMI	Mini HDMI
Boot	MicroSD	MicroSD	MicroSD	MicroSD
Red	Ethernet 10/100	-	Ethernet 10/100, Wifi, BT	Wifi y BT
Consumo	800mA / 4w / 5v	160mA / 0,8w / 5v	2,5A / 12,5w / 5v	160mA / 0,8w / 5v
Alimentación	MicroUSB / GPIO	MicroUSB / GPIO	MicroUSB / GPIO	MicroUSB / GPIO
Tamaño	85 x 56 mm	65 x 30 mm	85 x 56 mm	65 x 30 mm

Nota. P.E. (2015)

De los diseños mostrados se evidencia que las características de la placa con el modelo 3 B son superiores, por ende, se hará uso de dicha placa.

2.2.9 Algoritmos

La formación automática es una rama de la IA cuyo fin principal es establecer técnicas que permitan a las computadoras acceder a mayor información. Precisamente, implica la creación de programas que sean capaces de generalizar la conducta, el comportamiento a partir de la información proporcionada como ejemplos (IEEE Signal Processing Society, s.f.).

2.2.9.1 Tipos de algoritmos

Los algoritmos de aprendizaje automático se catalogan según su salida. 2 de los más relevantes, y que veremos después, serán el aprendizaje controlado y el aprendizaje no controlado (Domínguez Pavón, 2019). También, de estas 2 clasificaciones algorítmicas, tendremos otras, como el aprendizaje por refuerzo, la transmisión o la multitarea, que estarán fuera del alcance de este informe.

2.2.9.1.1 Aprendizaje Supervisado

El aprendizaje supervisado es un método que permite inferir una funcionalidad gracias a los datos conseguidos de las sesiones de capacitación. La información usada para la preparación consiste en vectores. En primer lugar, un elemento del par son los datos de entrada y, en segundo lugar, el resultado esperado (Stamm et. al, 2018).

Las salidas de las funciones podrían ser un valor numérico (vectorial o simple) o una etiqueta, para los inconvenientes de regresión y clasificación, respectivamente. El propósito de aprendizaje es, desde la observación de ciertos ejemplos, producir una función que realice predicciones sobre el valor que

corresponde a aquellos datos de entrenamiento. De esta forma, debe poder generalizar a partir de nueva información datos que no estén presentes en los vectores de entrenamiento (Stamm et.al, 2018).

De esta forma puesto que, los datos que conforman el proceso de formación constan de varios pares de muestras de entradas y salidas. Al conocer el desempeño, el sistema se favorece de la supervisión del maestro, de ahí el nombre de aprendizaje supervisado. Para un nuevo esquema de aprendizaje, en el paso (m+1) ésima, los pesos se ajustarán de la siguiente forma:

$$w_{ij}^{m+1} = w_{ij}^m + \Delta w_{ij}^m$$

2.2.9.1.2 Aprendizaje no Supervisado

En contraste al caso anterior, en este contexto no hay un conocimiento previo sobre las características del conjunto de datos. Por lo tanto, el aprendizaje no supervisado considera las entradas como un conjunto de variables aleatorias desconocidas, construyendo modelos de densidad para estos datos (Kaggle, 2017).

El santo grail de los aprendizajes no supervisados es la generación de códigos de factores a partir de datos, es decir, código con elementos estadísticamente independientes. El aprendizaje no supervisado genera superiores resultados una vez que los datos originales se traducen en códigos de factores. Buenos ejemplos son las reglas de aprendizaje de Hebb y las reglas de aprendizaje competitivo. Un primer ejemplo es una mejora de peso que conecta 2 nodos excitados simultáneamente.

Otra manera de este tipo de aprendizaje es la agrupación (clustering), que no se basa necesariamente en una distribución de probabilidad. De acuerdo

(Ramanath et.al, 2005), puede encontrarse una descripción de los diferentes algoritmos de agrupamiento y clasificación que utilizan en los aprendizajes no supervisados, con metodologías jerarquizadas y no jerárquicos. Estos métodos de agrupación jerárquica buscan construir un árbol de grupos basado en la distancia entre elementos. El siguiente ejemplo, nos ilustra de una manera muy simple estos métodos incluyen la siguiente matriz de distancias.

Figura 17

Matriz de distancias

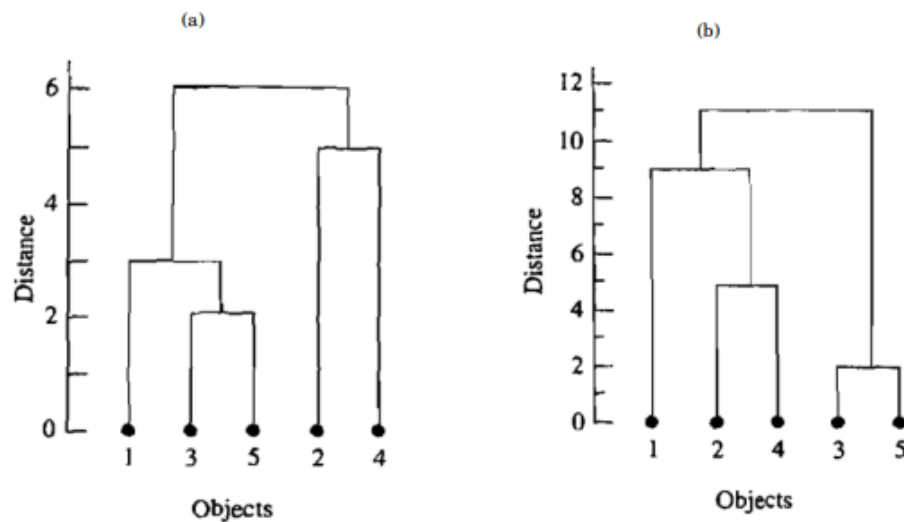
$$D = \{d_{ik}\} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & & & & \\ 9 & 0 & & & \\ 3 & 7 & 0 & & \\ 6 & 5 & 9 & 0 & \\ 11 & 10 & \textcircled{2} & 8 & 0 \end{bmatrix} \end{matrix}$$

Nota. “Identificación del modelo de cámara mediante Redes Neuronales Convolucionales” (Domínguez, 2019)

Implementando algoritmos de asociación simples y completos, tenemos la posibilidad trazar una imagen del grupo de elementos correspondiente a sus distancias mínima y máxima.

Figura 18

Resultados del diagrama de grupo de asociación simple y completo



Nota. “Identificación del modelo de cámara mediante Redes Neuronales Convolucionales” (Domínguez, 2019)

Como se puede ver en la Figura 19, ambos procedimientos generan resultados similares a pesar de las diferentes jerarquías de grupos, ya que uno trabaja de forma dinámica para la distancia máxima y el otro para la distancia mínima. Los elementos 1 y 3.5 en ambos algoritmos permanecen separados por un solo salto, pero la distancia entre ellos es de 3 unidades dependiendo de un algoritmo u otro.

Sin embargo, también existen algoritmos de agrupación en clústeres no jerárquicos. En este caso, no estamos tratando de encontrar relaciones entre los varios miembros de un grupo para ordenar, sino más bien de poder asociarlos en varios grupos. Uno de los algoritmos usados para esta unión ha sido planteado por MacQueen et.al (2000), quienes propusieron el término Kmeans para su descripción de tal manera que especifique cada elemento a ordenar en el conjunto de forma que la distancia al centro sea muy pequeña.

Este algoritmo se puede precisar d en los próximos 3 pasos:

1. Los elementos del grupo K se asignan inicialmente y los centroides de cada grupo o clusters se calculan.

2. La lista de elementos se considera asignando cada elemento al grupo con el centro más cercano. Por lo general, esta distancia es calculada utilizando la distancia euclidiana con observaciones normalizadas (estandarizadas) o no normalizadas. Se volverá a calcular el centroide del clúster que recibe el nuevo elemento, así como el elemento perdido.

3. Se vuelve a desarrollar el paso 2 hasta que no sea posible realizar más reasignaciones.

Una variante consta en la asignación del centroide K especificado al principio y podemos ir directamente al paso 2. Como podemos ver, este es un algoritmo muy simple y fácil de programar. Se pueden encontrar más ejemplos prácticos de este algoritmo en el libro de (Ramanath et.al, 2005).

2.2.9.1.3 Aprendizaje profundo

Aprendizaje Profundo (Deep Learning) se le denomina así, a un grupo de algoritmos que buscan modelar abstracciones con un elevado nivel de datos utilizando estructuras compuestas por muchas transformaciones no lineales.

Cada observación se puede representar de diferentes maneras, sin embargo, varias de ellas facilitan el aprendizaje de tareas de interés a partir de ejemplos e investigaciones del campo tratando, e intenta determinar cuál es buena y cómo modelarlas para reconocerlas.

Al centrarse en el dominio computacional, que es de mayor interés para esta memoria, tienen la posibilidad de descubrir varias construcciones de

aprendizaje profundo (redes neuronales integradas y profundas). En estas aplicaciones, estas redes han demostrado buenos resultados.

De forma general, hablamos de un grupo de algoritmos que fueron generados para un aprendizaje de manera automática. Iniciando de este punto, hay diferentes propiedades y características, tales como:

- Emplear una estructura de capas en cascada que consiste en unidades no lineales, tanto para extraer como para transformar las variables. Cada capa toma la salida de la capa precedente como su entrada. Los algoritmos empleados tienen la posibilidad de usarse tanto para el aprendizaje supervisado y el no supervisado. Su aplicación, entre las más comunes integran el modelamiento de datos y el reconocimiento de patrones.
- Está basado en el aprendizaje de la presentación de datos o diversos niveles de propiedades. La jerarquización se genera derivando las propiedades de máximo nivel desde las de más bajo nivel.
- Aprender la representación en sus diversos niveles, en base de los distintos niveles de abstracción. Estos niveles se clasifican conformando una jerarquía de conceptos.

Cada una de estas maneras de definición del aprendizaje profundo poseen en común:

- Para su procesamiento, hay varias capas no lineales.
- Los aprendizajes supervisados y no supervisado, de representaciones de características en cada capa

2.2.10 Lenguajes de programación

Existen diversas posibilidades para llevar a cabo los algoritmos de aprendizaje de manera automática. Cabe resaltar que, dado al enorme crecimiento y mejoramiento en este campo, genero el surgimiento de diversos programas y lenguajes de programación. Los cuales puede que ya se encuentre obsoleta en el momento de finalizar la memoria.

2.2.10.1 Programación Python

Python es una forma de lenguaje de programación empleado en las diversas aplicaciones computarizadas, muy flexible, sencillo y fácil de aprender. Python no sólo es multiplataforma y multiparadigma, sino que también nos sirve para plasmar cualquier tipo de vía. El lenguaje de programación posee un frameworks de gran calibre, los cuales favorecen y ayudan al desarrollo web.

2.2.11 Librerías de programación

2.2.11.1 OpenCV

OpenCV, que significa Open Source Computer Vision, es una versátil biblioteca originalmente concebida por Intel para brindar soporte a los primeros compiladores Intel C++ y Microsoft Visual C++ en arquitecturas x86. Actualmente, esta librería desempeña un papel fundamental en el procesamiento de imágenes y la visión artificial. Su aplicabilidad se extiende a una amplia gama de campos, incluyendo supervisión de procesos, sistemas de seguridad con detección de desplazamiento, identificación de objetos y robótica avanzada. Según la página oficial de OpenCV, esta es una librería de código abierto y de uso libre distribuida bajo licencia BSD, lo que la hace accesible para proyectos comerciales e investigativos.

Aunque inicialmente perfeccionada para procesadores Intel, OpenCV es versátil y puede aprovechar procesadores de diferentes tipos, incluso multinúcleo. Además, es compatible con sistemas operativos como GNU/Linux, Windows, Android, iOS y Mac OS X. Es relevante destacar que OpenCV también ha experimentado un crecimiento dinámico en interfaces para Python, Ruby, Matlab y otros lenguajes (Bradski y Kaehler, 2008)

La estructura de OpenCV consta de cinco componentes principales:

- CV: Contiene algoritmos desde los conceptos fundamentales hasta los más sofisticados en el procesamiento de imágenes y la visión por computadora
- ML (Machine Learning): Ofrece una biblioteca de aprendizaje automático con una variedad de clasificadores estadísticos y herramientas de agrupación.
- HighGUI: Cuenta con funciones de entrada/salida para la gestión de video e imágenes.
- CXCore: Incluye estructuras básicas, algoritmos, soporte XML y funciones gráficas. Recibe aportes de CV, MIL y HighGUI.
- CvAux: Cuenta con algoritmos experimentales.

En cuanto al procesamiento de imágenes, OpenCV se esfuerza por proveer un soporte autorizado y sencillo. La biblioteca ofrece una amplia gama de funciones empleadas en diversas aplicaciones de visión por computadora, incluyendo la supervisión de productos, reconocimiento de personas u objetos en desplazamiento, identificando y distinguiendo rostros humanos, imágenes

médicas, interfaces de usuario, seguridad, reconstrucción 3D y robótica, entre otros. Su adaptabilidad y versatilidad la convierten en una herramienta esencial en el ámbito de la visión por computadora.

2.2.11.2 Numpy

NumPy es una destacada biblioteca de Python para realizar cálculos científicos, cuyo nombre es una abreviatura de "Python numérico". La descripción oficial de esta biblioteca, que se puede localizar en su página de internet, es la siguiente:

"NumPy se erige como el pilar esencial en el ámbito de la computación científica con Python. Entre sus componentes, se destacan:

- Un potente objeto de arreglo N-dimensional.
- Tareas de radiodifusión sofisticadas.
- Implementos para la incorporación de código en C/C++ y Fortran.
- Capacidad para realizar álgebra lineal, transformadas de Fourier y generación de números aleatorios de manera eficiente.

Además de sus aplicaciones evidentes en el campo científico, NumPy puede desempeñar el papel de un contenedor multidimensional eficaz para datos genéricos, permitiendo la definición de tipos de datos arbitrarios. Esto posibilita una integración sin inconvenientes con una extensa diversidad de banco de datos.

NumPy se distribuye bajo la licencia BSD, lo que lo hace altamente versátil y reutilizable, con mínimas restricciones".

NumPy es tan fundamental en el ecosistema de Python que varias bibliotecas, incluyendo pandas, se construyen completamente sobre su base.

2.2.11.3 MediaPipe

Es un conjunto de herramientas y bibliotecas desarrolladas por Google que se centra en la percepción y el seguimiento de objetos en medios visuales, como imágenes y videos. Ofrece una diversidad de capacidades, incluido la detección de rostros, monitoreo de manos, identificación y registro de gestos y más, lo que permite a los desarrolladores incorporar fácilmente funcionalidades de visión por computadora en aplicaciones y proyectos. MediaPipe se destaca por su capacidad de funcionar en el tiempo instantáneo en una extensa gama de dispositivos y plataformas, lo que lo hace adecuado para aplicaciones de realidad aumentada, realidad virtual, interacción natural con el usuario y muchas otras áreas relacionadas con el procesamiento de imágenes y videos. (Lugaresi et.al, 2019)

2.2.12 Kali Linux

Kali Linux pertenece a Linux y esta hecha para realizar pruebas y auditorias de seguridad. Anteriormente se conocía como BackTrack y se utiliza en seguridad informática y hacking ético. Los propósitos de Kali Linux son:

- Pruebas de penetración: Proporciona herramientas y programas para evaluar la seguridad de sistemas y redes. Se realizan simulaciones de ataques para descubrir fallas en el sistema.
- Hacking ético: Ofrece un entorno completo de herramientas especializadas para detectar debilidades de seguridad.

- Formación de seguridad: Kali Linux sirve para aprender sobre técnicas de hacking ético y medidas preventivas para seguridad.
- Desarrollo y pruebas de seguridad: Identifica y corrige posibles vulnerabilidades ante ataques malintencionados.

Figura 19*Kali Linux*

Nota. Recuperado de: (Kali Linux, 2024)

2.2.13 Ubuntu MATE

Ubuntu es una distribución de LINUX, en base de la arquitectura de Debian. Ahora funciona en computadoras de mesa y servidores en línea, en sistemas Intel, AMD y ARM. Esta patrocinado por Canonical y es financiado por servicios del sistema operativo y ofertando servicios técnicos. Al ser un sistema libre, se aprovecha el talento de otros desarrolladores para mejorar sus componentes del sistema operativo. Adicionalmente, la comunidad desarrolladora brinda soporte para otros entornos gráficos como: (Rodolfo Perez, 2020)

- Xubuntu

- Kubuntu
- Edubuntu
- Lubuntu
- Ubuntu MATE

Página oficial de Ubuntu: <http://www.ubuntu.com/>

Figura 20

Escritorio de UbuntuMATE



Nota. Obtenido de: (Ubuntu, 2023).

2.2.14 RISC OS

Reduced Instruction Set Computer (RISC OS) tiene 3 componentes principales:

1. **Núcleo:** Contiene los programas que hacen funcionar la computadora.
2. **Interfaz de línea de comandos:** Es una interfaz basada en texto para escribir programas y scripts.

3. Escritorio: Se utiliza íconos que representan archivos, directorios y aplicaciones. Además, ejecutan aplicaciones en Windows. Se puede tener muchas ventanas en simultáneo (RISC OS Open Ltd, 2018).

RISC OS es el nombre del sistema operativo de la computadora. Además, se encarga de controlar la forma en que aparece la pantalla de la computadora y sus funciones y tareas.

El escritorio es el área de trabajo. La barra en la parte inferior de la pantalla. Mientras se realiza las tareas y se usan aplicaciones, se agregarán más tareas al escritorio. El escritorio es una interfaz gráfica de usuario que utiliza Windows, íconos, menús y punteros.

Figura 21

Interfaz de escritorio



Nota. Recuperado de: (RISC OS Open Ltd, 2018).

2.3 ESQUEMA CONCEPTUAL

Se muestra en la Figura 22, el principio de funcionamiento del sistema detector de EPP. Donde en el primer cuadro de la parte izquierda se muestra al personal con los implementos de protección.

Seguidamente en la tabla 1 se muestran cámaras, las cuales son responsables de la captura de fotografías de diferentes ángulos, usando cámara web 2K Full HD. Con una resolución de 2560x1440. Además, están de pequeño tamaño y enfoque automático.

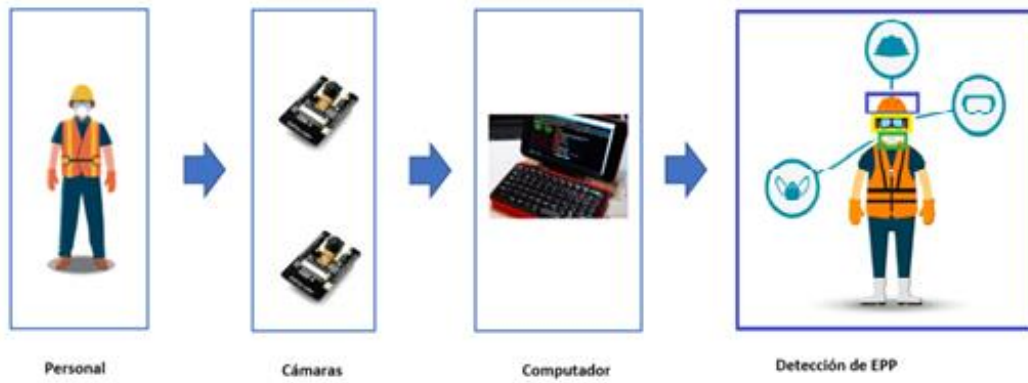
El computador Raspberry Pi tomará el mayor trabajo, ya que dentro de este se realizará el procesamiento y tratamiento de la data. En esta fase se elabora el aprendizaje máquina de reconocimiento o diferenciación de los elementos de resguardo personal.

Finalmente se muestra por una pantalla o monitor, si verdaderamente el trabajador está usando los EPP. Este será capaz de indicar en cuadros de colores con sus respectivos nombres los EPP captados.

Con este sistema se busca supervisar de forma inteligente, si el personal trabajador está cumpliendo con el uso de los EPP. Este sistema además busca facilitar a la empresa con la seguridad de sus trabajadores y de forma psicológica el trabajador trate de cumplir las respectivas medidas.

Figura 22

Diagrama de funcionamiento del detector de EPP



Personal

Cámaras

Computador

Detección de EPP



CAPITULO III: ASPECTOS GENERALES, METODOLOGÍA Y DESARROLLO

3.1 ASPECTOS METODOLÓGICOS

3.1.1 Estrategia

Las pruebas se realizarán con los EPP más cercanos al rostro como son los cascos, cubrebocas y lentes de seguridad, esto porque estamos usando un algoritmo de clasificador de reconocimiento facial. También se colocará las 2 cámaras en diferentes ángulos, para que la captura o toma sea más precisa, respecto a la luminosidad del ambiente.

En la parte de software se tratará de adaptar y usar los lenguajes de programación más potentes y livianos, como es el lenguaje de Python. Esto precisamente para evitar la congestión en el procesamiento de datos.

3.1.2 Tipo de investigación

Cuasiexperimental:

- La finalidad es detectar los EPP que lleva el trabajador.
- Se busca que el sistema de detección de EPP tenga una respuesta rápida.
- Las pruebas del sistema de detección con el medio y la luminosidad deben de ser óptimo.

3.1.3 Nivel de investigación

Aplicativo:

Propone solucionar los problemas de seguridad con el cumplimiento de los EPP. Además, se pone en práctica estos nuevos sistemas de aprendizaje máquina.

En esta tesis se enmarcará la innovación tecnológica, desplazando al clásico sistema de observación. Para ello identificamos los indicadores como es el uso de casco, lentes de seguridad y mascarilla.

Se usarán los conocimientos existentes, con respecto a temas de aprendizaje máquina, sistema de visión artificial y el aprendizaje profundo. Tomando en cuenta la electrónica, concierne a la transferencia de la data y el procesamiento de imágenes.

3.1.4 Métodos

En este capítulo se manifiesta la metodología utilizada en la investigación, donde expondrá la elaboración y como se obtendrá el diseño del prototipo, utilizando los siguientes métodos:

A. Método bibliográfico o documental

Se aplica este método porque Raspberry pi es considerada una tecnología nueva, por ello es necesario acrecentar nuestros conocimientos sobre dicha tecnología, sobre los implementos que necesita y su proyección de empleo en los diversos proyectos que se presentan en nuestra sociedad, siempre buscando mejorar, modernizar y sobre todo beneficiar a la sociedad y el medio que nos rodea. Esta investigación se respalda en una variedad de documentos y emplea los métodos que se detallan a continuación:

- **Investigación Bibliográfica**

Esta exploración se reconoce como la fase inicial dentro de un proceso investigador que proporcionará conocimiento de manera sistemática, empleando distintos recursos para llevar a cabo una gran búsqueda de información bibliográfica, conocimientos y prácticas sobre un tema específico.

La pesquisa, en la literatura, constituye una manera de empezar el texto (introducción) en cualquier tipo o clase de investigación, o como una etapa fundamental en su inicio, ya que proporciona todo el conocimiento disponible de investigaciones previas, ofreciendo al investigador teorías, conceptos, experimentos, herramientas, hipótesis, resultados y métodos empleados sobre el tema de estudio propuesto.

La correcta utilización de este tipo de exploración dependerá de diversas habilidades como investigación, toma de apuntes, selección y evaluación de materiales. Las conclusiones que se obtengan mediante este método, deberá elaborar un informe adecuado a la investigación.

Este método termina al lograr el objetivo de la investigación: la redacción y ceración de un artículo científico que abarque todos los datos recolectados en el proceso, desde los problemas hasta las conclusiones derivados de la investigación.

- **Investigación Hemerográfica**

Este tipo de exploración se distingue por examinar documentos científicos, así como ensayos de revistas y periódicos, con el propósito de obtener una cantidad significativa de información relevante para el estudio en curso.

Usualmente, se realizaba en una biblioteca de periódicos donde se consultaban publicaciones, sin embargo, gracias a los avances tecnológicos, hoy es posible hallar toda la información en el internet, debiendo tener precaución con el contenido, pues la información recabada podría ser falsa; de ahí la aplicación de diversas técnicas para asegurar que la información provenga de fuentes reales y confiables.

Esta forma de investigación debe ser vista dentro del marco de las tendencias más actuales, lo que implica buscar ubicaciones que reúnan colecciones organizadas de diversas publicaciones periódicas, como periódicos, diarios y revistas.

En esta investigación es esencial identificar claramente la temática y ofrecer referencias precisas de las publicaciones para evitar confusiones. En otros casos, es necesario definir con claridad el propósito y objetivo del estudio.

3.2 CAMPO DE VERIFICACIÓN

3.2.1 Ubicación espacial:

El lugar de pruebas y la implementación momentánea, se realizará en una habitación común simulando el área de trabajo de una industria.

3.2.2 Ubicación Temporal:

El sistema desarrollado de visión artificial está usando tecnologías y métodos actuales, aplicado a la detección de EPP en tiempo real.

3.2.3 Unidades de Estudio:

El estudio se realizará al reconocimiento de los EPP que usa un supuesto trabajador dentro de una empresa industrial. Y se evaluará las pruebas con el cumplimiento de uso de los EPP en los trabajadores.

3.3 TÉCNICAS E INSTRUMENTOS

3.3.1 Técnicas.

La técnica a emplear para la ejecución de este proyecto, es análisis y la respuesta del algoritmo de aprendizaje máquina a la toma de fotografías o video por las cámaras, y éstas serán procesadas posteriormente con cuadros de colores, y donde se indique si se cumple o no los EPP.

3.3.2 Instrumentos

No podemos especificar el instrumento de medida, pero si el tiempo de respuesta. Además, se evaluará la viabilidad del proyecto, con el alcance de solución al cumplimiento de los EPP en Industrias donde el uso, es obligatorio.

Los materiales y equipos a usar son pocos ya que se enfoca más a la parte de software, seguidamente se muestra el alcance de los principales materiales, que se usarán en esta tesis.

3.3.2.1 Materiales

- **Cámara:** Es la herramienta de adquisición de fotografía e imagen de diferentes ángulos del personal de trabajo industrial.

Tabla 1

Cantidad de Cámaras

Materiales	Cantidad	Modelo
Cámara	2	HD-WEBCAM

- **Ordenador:** La Raspberry Pi es una gama de computadoras de bajo coste, también conocidas como computadoras de placa única o de placa reducida, creadas en el Reino Unido por la Raspberry Pi Foundation. Su propósito principal es democratizar el acceso a la informática y la creación digital a nivel global.

Tabla 2

Cantidad de Ordenadores

Materiales	Cantidad	Modelo
Ordenador	1	Raspberry pi 3

- **EEP:** Se usarán los principales equipos de protección personal, además estos deben de ser de un tamaño medianamente grande y fácil de reconocer por nuestro algoritmo a crear.

Tabla 3

Cantidad de Epp

Materiales	Cantidad	Modelo
Casco	1	Amarillo
Lentes	1	De visibilidad
Mascarilla (Respirador)	1	MSA
Total elementos	3	

3.4 ASPECTOS GENERALES DE INSTRUMENTOS UTILIZADOS

3.4

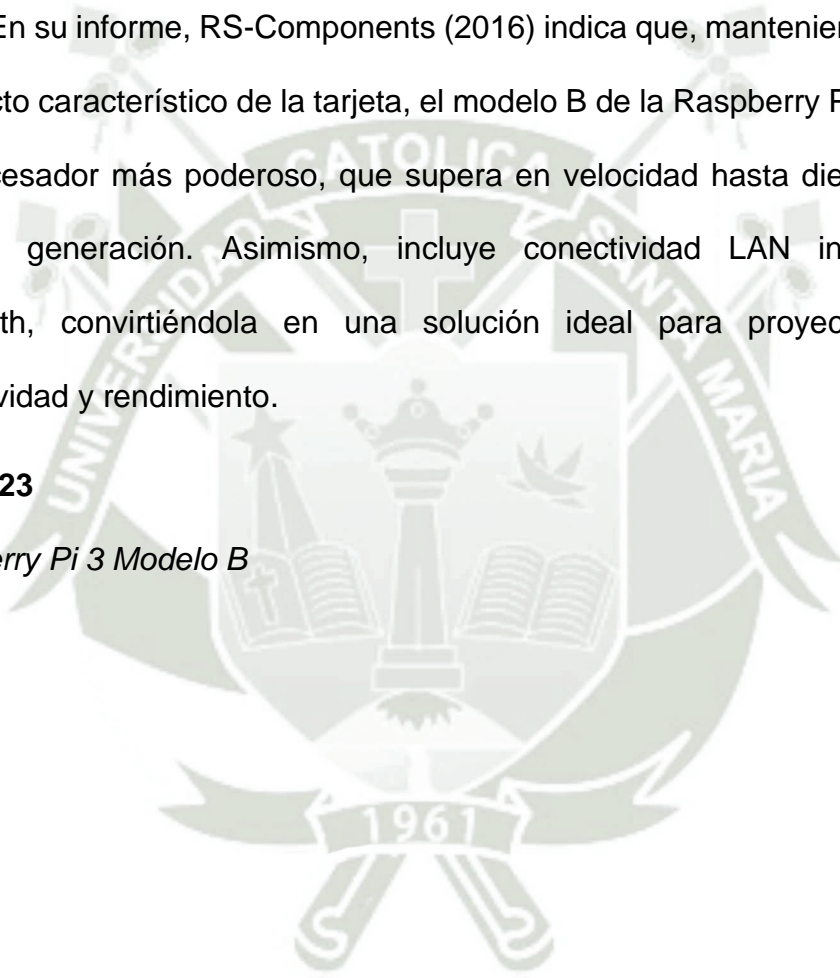
3.4.1 RASPBERRY PI 3

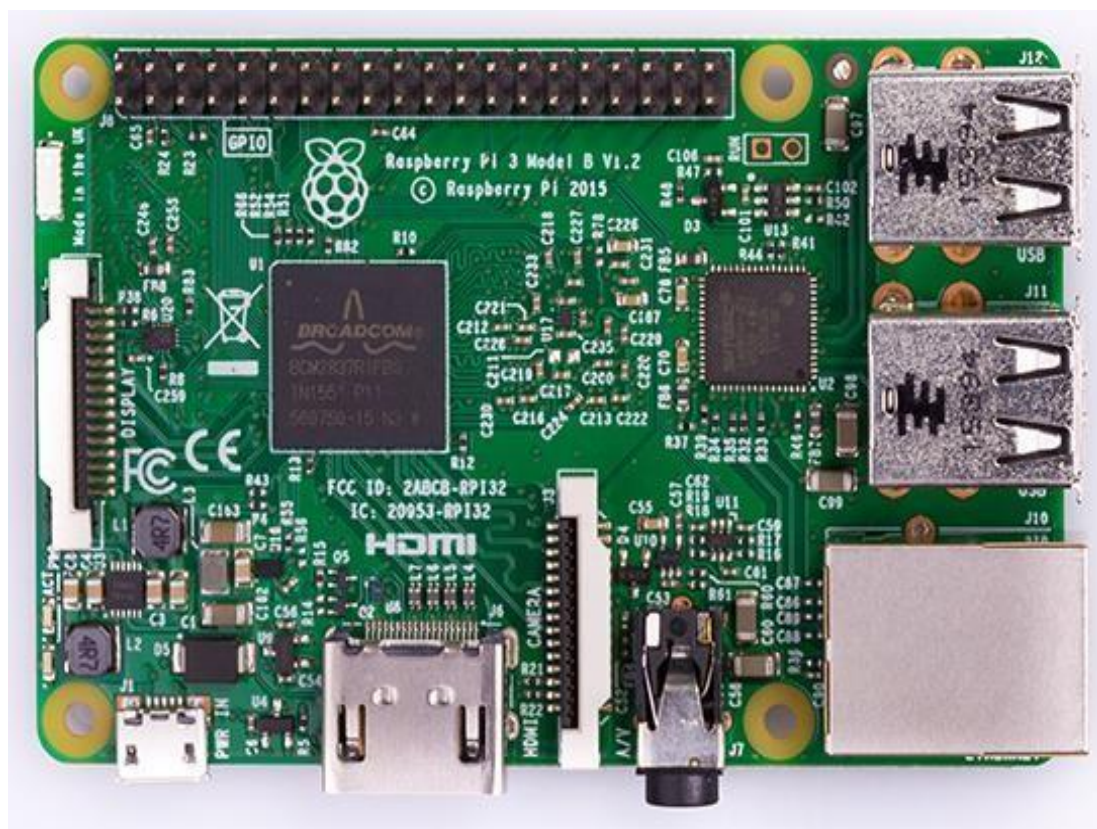
RS-Components (2016) describe la Raspberry Pi 3 como una placa computacional resistente, de dimensión similar a una tarjeta de crédito, con una amplia gama de usos y la capacidad de reemplazar fácilmente a modelos anteriores, como la Raspberry Pi B+ y la Raspberry Pi 2.

En su informe, RS-Components (2016) indica que, manteniendo el diseño compacto característico de la tarjeta, el modelo B de la Raspberry Pi 3 incorpora un procesador más poderoso, que supera en velocidad hasta diez veces a la primera generación. Asimismo, incluye conectividad LAN inalámbrica y Bluetooth, convirtiéndola en una solución ideal para proyectos de alta conectividad y rendimiento.

Figura 23

Raspberry Pi 3 Modelo B



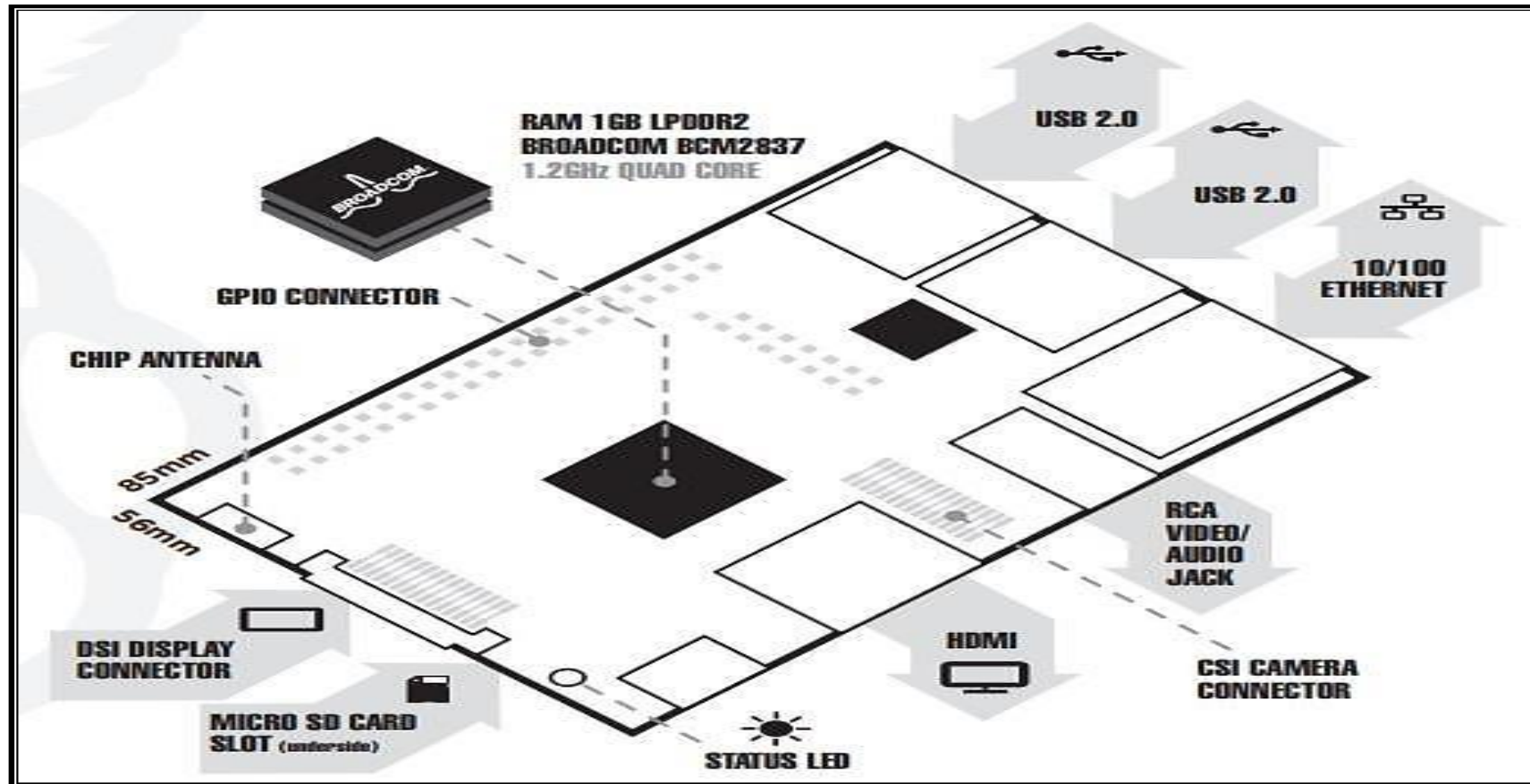


Nota. Raspbian (2021)



Figura 24

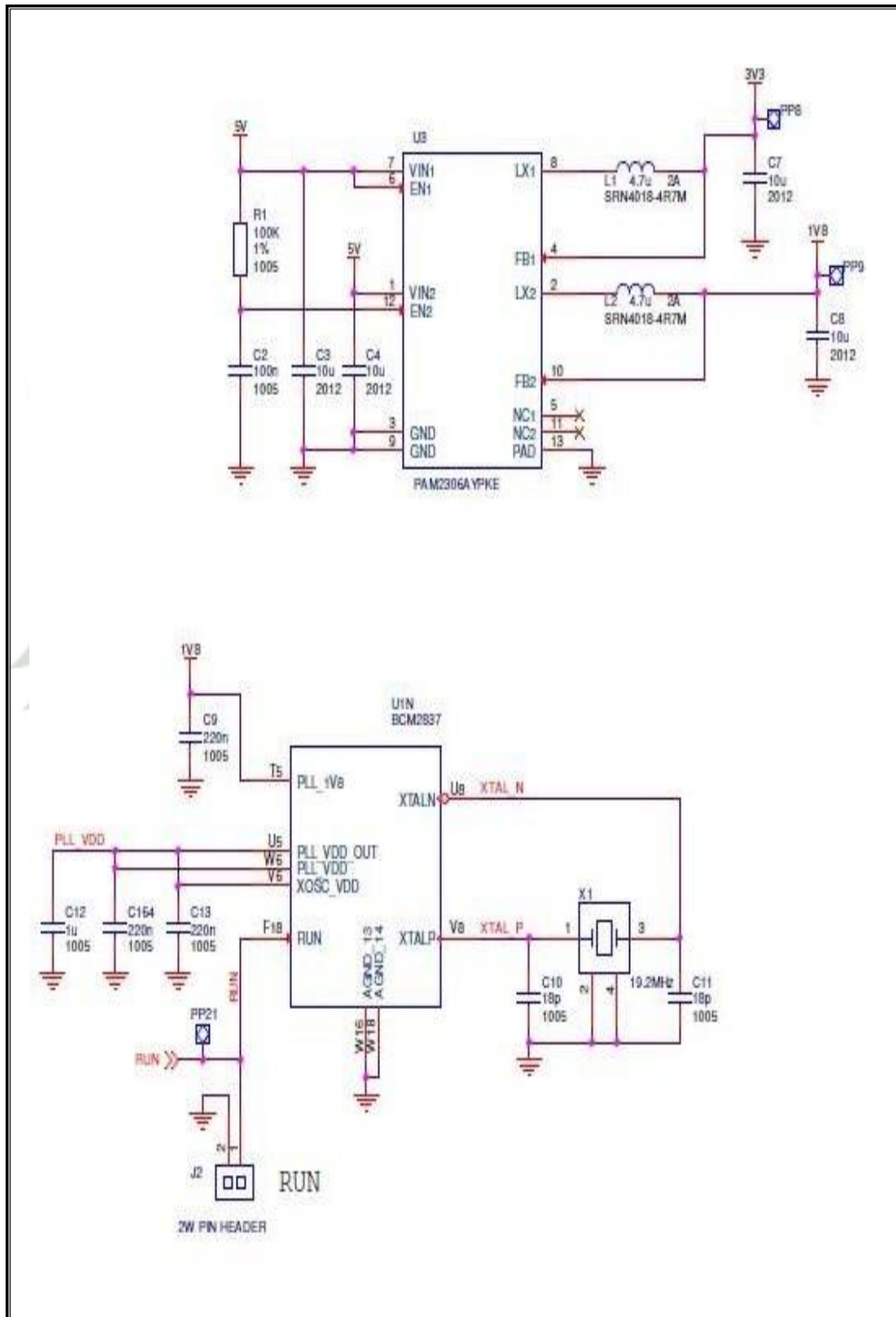
Componentes De Una Raspberry Pi 3



Nota. Gonzales (2015)

Figura 25

Esquemático Raspberry Pi 3 (B)



Nota. Alasdair (2022)

3.4.1.1 Características de Raspberry Pi 3

Tabla 4

Especificaciones De Raspberry Pi 3

SoC	Broadcom BCM2387 chipset
CPU	1.2GHz Quad-Core ARM Cortex-A53
GPU	Dual Core Video Core IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated Open VG, and 1080p 30 H.264 high-profile decode.
RAM	1GB LPDDR2
Sistema Operativo	Boots desde la tarjeta Micro SD, ejecutando una versión del sistema operativo Linux o Windows 10 IoT
Dimensión	85 x 56 x 17mm
Alimentación	Micro USB socket 5V1, 2.5A

Nota. Obtenido de: (Pastor, 2018)

Tabla 5

Conectores De Raspberry Pi 3

Ethernet	10/100 BaseT Ethernet socket
Video	HDMI (rev 1.3 & 1.4 Composite RCA (PAL and NTSC)
Audio	Audio Output 3.5mm jack, HDMI
USB	USB 4 x USB 2.0 Connector
GPIO	40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
Conector de cámara	15-pin MIPI Camera Serial Interface (CSI-2)
Conector de pantalla	Display Serial Interface (DSI) 15-way flat flex cable connector with two data lanes and a clock lane
Slot de memoria	Push/pull Micro SDIO

Nota. Obtenido de: (Pastor, 2018)

3.4.2 Hardware Broadcom BCM2837

El SoC emplea Broadcom BCM2837 como su unidad de procesamiento gráfico o GPU y microprocesador. La empresa de Broadcom es conocido como un proveedor sin fábrica propia (fabless), proporcionando diseños de productos como Propiedad Intelectual (PI), mientras otras empresas se encargan de fabricar los chips de silicio.

La sección BCM2837 del propio SoC presenta un microprocesador Quad-Core ARM Cortex-A53 que básicamente opera a 1.2 GHz y una GPU Broadcom VideoCore® IV. Asimismo, el BCM2837 es un diseño para móviles, y se enfoca en minimizar el consumo de energía para maximizar la vida útil de la batería. Su velocidad de reloj relativamente baja ayuda en gran parte a reducir el consumo de energía, lo que permite que funcione a 1.2 GHz. De tal forma que esta baja velocidad también implica un voltaje menor, pues disminuye el calor y prolonga su durabilidad del chip.

Aunque es posible potenciar el BCM2837, conocido como overclocking, esto generalmente no se recomienda, ya que puede hacer que el microprocesador sea menos estable y reducir su vida útil.

La GPU Broadcom VideoCore IV administra el procesamiento de audio y video para el SoC. Pues, según Donald Norris (2014), es totalmente compatible con el estándar Open GL ES 2.0, porque es capaz de ejecutar esta interfaz de programa de aplicación (API) en hardware incrustado, como el Broadcom 2837. En resumen, esto permite al 2837 mostrar gráficos 3D y ejecutar video HD utilizando shaders y filtros de textura necesarios para juegos modernos. Además,

este chip admite la decodificación de alto perfil de H.264 a 1080p y 30 cuadros por segundo.

Figura 26

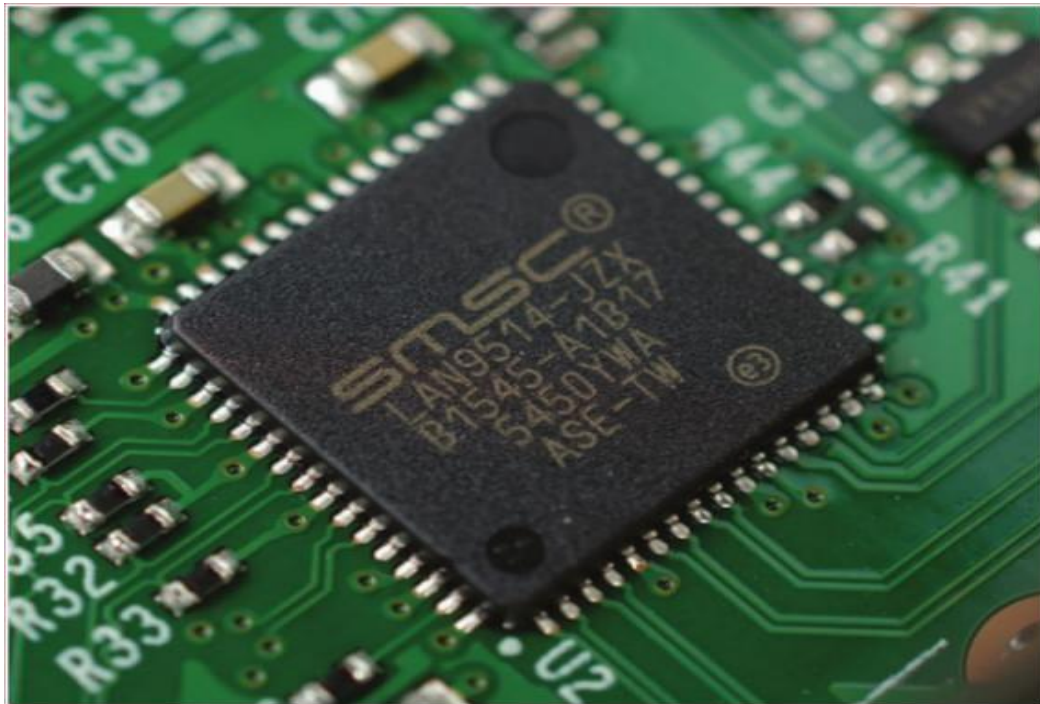
Broadcom Bcm2837.



Nota. Russell (2016)

3.4.3 USB Chip

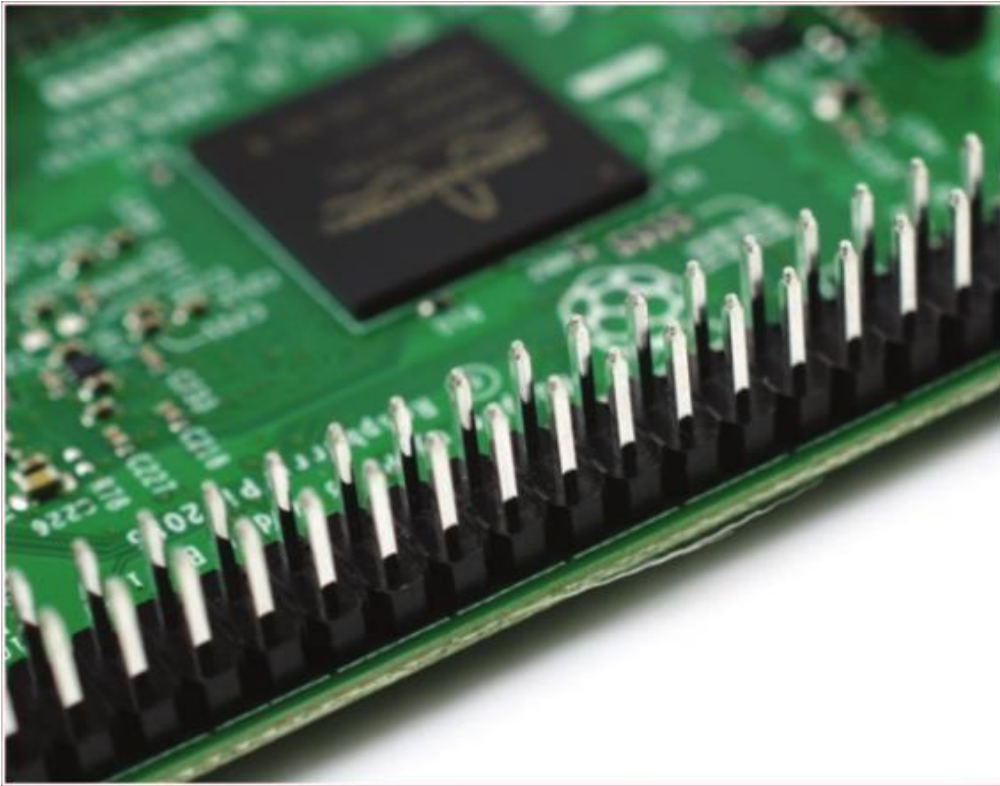
Conforme a Raspberry (2016), el Raspberry Pi 3 comparte el mismo componente SMSC LAN9514 que su antecesor, el Raspberry Pi 2, incorporando conectividad Ethernet y cuatro puertos USB a la placa. De manera similar, el componente SMSC se enlaza al SoC mediante un único puerto USB, funcionando como un adaptador USB-Ethernet y un concentrador USB.

Figura 27*Usb Chip*

Nota. Russell (2016)

3.4.4 GPIO

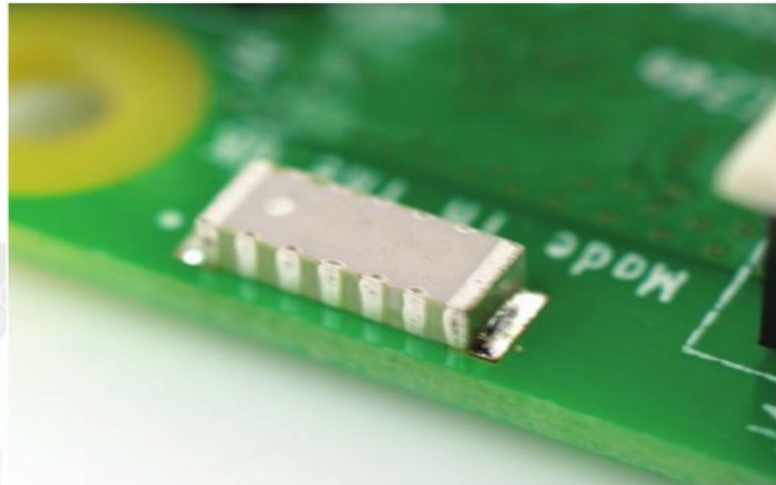
La Raspberry Pi 3 presenta un conector de entrada y salida de 40 pines de propósito general (GPIO) que son básicamente el mismo, pues todos tienen su origen en el modelo B+ y en el modelo A+. De acuerdo con la Fundación Raspberry Pi (2016), cualquier hardware GPIO existente operará sin alteraciones; la única modificación es un interruptor al que se expone UART en los pines del GPIO, pero que es controlado por el sistema operativo internamente.

Figura 28*Gpio.*

Nota. Russell (2016)

3.4.5 Antena

La Fundación Raspberry Pi sostiene que no es necesario enlazar la Raspberry Pi 3 a una antena externa. Por su parte, Raspberry Pi Foundation (2016) afirma: "Sus radios están vinculadas a esta antena soldada directamente a la placa, con el propósito de mantener el tamaño del dispositivo al mínimo" (párr. 7). Aunque su tamaño es pequeño, esta antena debería ser completamente capaz de captar señales inalámbricas Bluetooth y LAN de manera efectiva, incluso a través de obstáculos como paredes.

Figura 29*Antena*

Nota. Russell (2016)

3.4.6 Memorias

Según Donald Norris (2014), la Raspberry Pi opera básicamente una memoria flash (SD) y memoria (DRAM). Asimismo, la versión original (modelo A), contaba con 256 MB de RAM, mientras que el más reciente (Modelo B), posee 1 GB. Disponer de 1 GB de DRAM asegura un funcionamiento eficiente del sistema y la ejecución correcta de programas, siempre y cuando se configuren de manera apropiada.

En cuanto a la memoria (SD), el señor Donald Norris (2014) indica que se utiliza para almacenar programas y datos así como el sistema operativo que requieran persistencia. Además, se brinda la posibilidad de elegir la capacidad de la tarjeta SD, la cual se introduce fácilmente la parte posterior de la placa.

Las tarjetas SD están clasificadas por clase, lo que indica la velocidad mínima de transferencia de datos que pueden manejar. Dado que la tarjeta SD actúa como un disco duro, una mayor velocidad es preferible (Norris, 2014).

Tabla

6

Designaciones De Clase De Tarjetas SD

Clase	Rendimiento mínimo	Unidades
Clase 2	2	MB/sec
Clase 4	4	MB/sec
Clase 6	6	MB/sec
Clase 10	10	MB/sec

Nota. Obtenido de: Libro Raspberry Pi Projects For The Evil Genius

3.4.7 Conector de alimentación

Tipo micro USB, diseñado para suministrar líneas de corriente continua de 5 voltios (V) y 2.5 amperios (A). No hay conexiones de datos vinculadas a este enchufe y puede emplear cualquier tipo de cargador de smartphone que posea un conector micro USB.

Figura 30

Conector De Alimentación



Nota. Raspberrypi.org (2022)

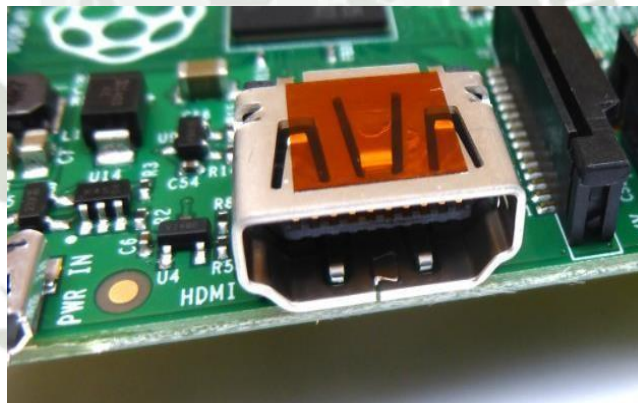
3.4.8 Conector HDMI

La Raspberry Pi brinda vídeo y sonido a través de un puerto HDMI compatible con casi todos los estándares.

El Raspberry Pi puede desempeñar el papel de un control remoto avanzado para dispositivos compatibles con HDMI-CEC, donde CEC proviene de Consumer Electronics Control. Este protocolo de un solo hilo y bidireccional se emplea para controlar dispositivos de audio y vídeo. HDMI-CEC ha sido adoptado por distintos fabricantes de equipos audiovisuales, como Sony con su Sharp con Aquos Link, BraviaLink, LG con Simplink, , Samsung con Anynet+, etc.

Figura 31

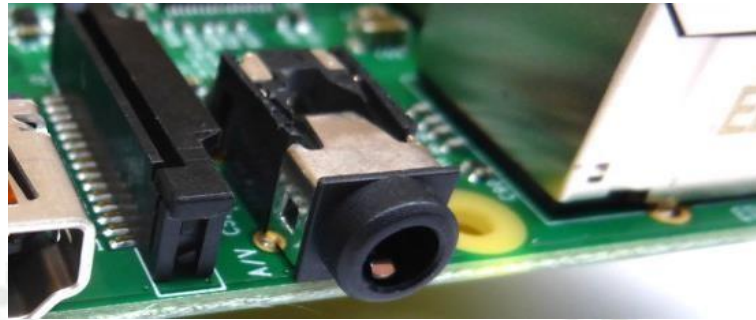
Conector HDMI



Nota. Raspberrypi.org (2022)

3.4.9 Conector de audio

La Raspberry Pi cuenta con conector estéreo estándar de 3,5 mm (salida de audio). Este audio comúnmente representa el equivalente analógico del audio digital transmitido desde el conector HDMI. Será necesario utilizar un amplificador de audio para disfrutar de la música, ya que no produce una señal lo bastante potente para alimentar un altavoz no amplificado.

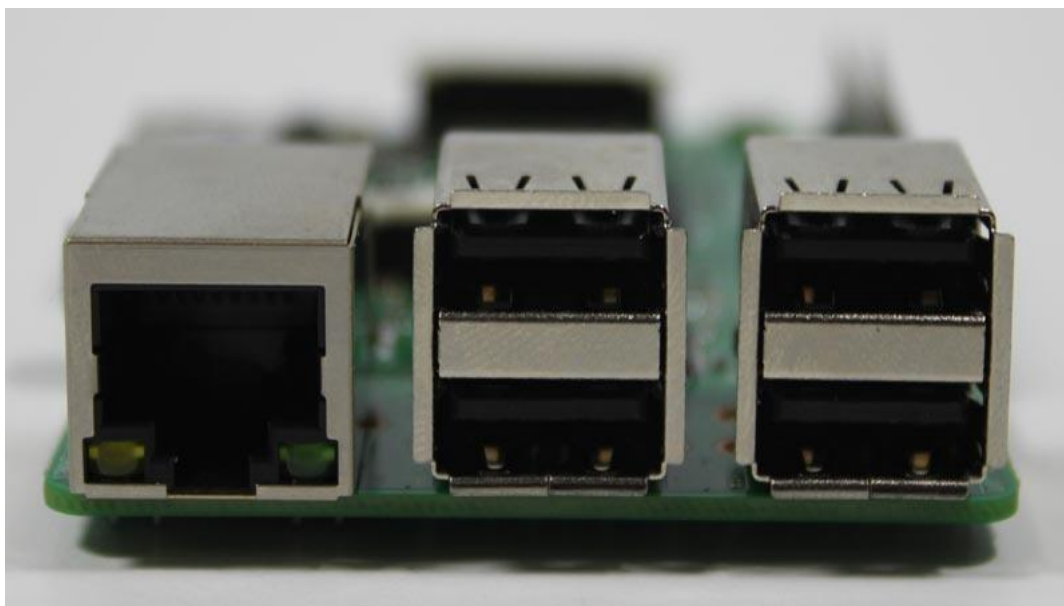
Figura 32*Conector De Audio*

Nota. Raspberrypi.org (2022)

3.4.10 Conectores Ethernet y USB

El puerto Ethernet utiliza un conector RJ45; simplemente se conecta el cable Ethernet desde este zócalo al enrutador o conmutador. La Raspberry Pi llevará a cabo automáticamente una "negociación" con el enrutador para obtener una dirección IP a través del proceso DHCP (Protocolo de Configuración Dinámica de Host).

Por otro lado, la placa Raspberry Pi cuenta con cuatro puertos USB que detectan los dispositivos USB conectados a ellos. Estos puertos están distribuidos en dos conjuntos de dos puertos USB, y tienen la misma función de reconocer los periféricos conectados a la placa a través de los zócalos USB.

Figura 33*Conectores Ethernet Y Usb*

Nota. Raspberrypi.org (2022)

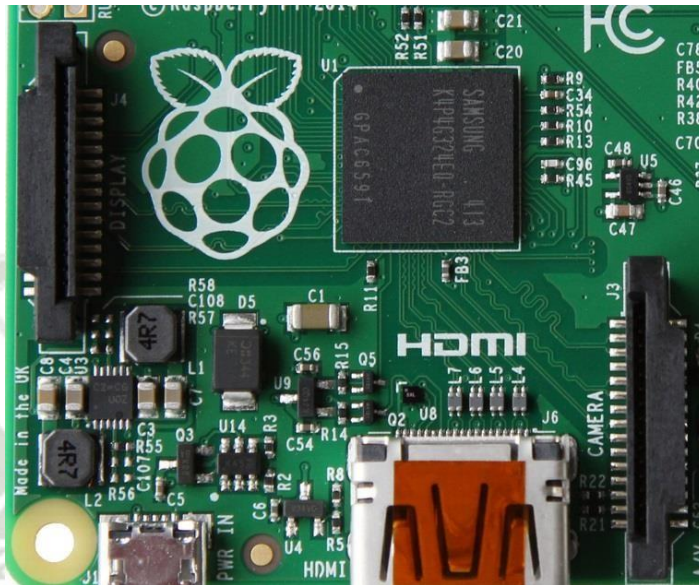
3.4.11 Conectores de cámara y display

El conector de Ethernet se compone de un conector RJ45; y cuando se estos tienen la función de reconocer los periféricos USB enchufados en los zócalos. Conecte su cable de conexión Ethernet en el puerto y el otro extremo al enrutador o al conmutador, la Raspberry Pi negociará automáticamente para obtener una dirección IP (Protocolo de Internet) en un proceso conocido como DHCP (Protocolo de Configuración Dinámica de Host).

La Raspberry Pi cuenta con cuatro puertos USB; estos puertos reconocen los periféricos USB conectados a los zócalos de la placa. La Raspberry Pi posee dos hileras de dos puertos USB, estos tienen el propósito de detectar los dispositivos USB conectados a los puertos correspondientes.

Figura 34

Conectores De Cámara Y Display.



Nota. Raspberrypi.org (2022)

3.4.12 Software

"Raspberry Pi fue concebido para funcionar en un entorno Linux", señala Norris (2014). La elección de diseño de la Raspberry Pi difiere notablemente de otras placas de microcontrolador similares, como la reconocida serie Arduino, que sigue un enfoque diferente. Esto no indica que una placa Arduino sea menos efectiva que una Raspberry Pi; simplemente muestra que el uso de la Raspberry Pi proporciona más flexibilidad y capacidades adicionales a los proyectos gracias a su entorno Linux.

3.4.13 Raspbian

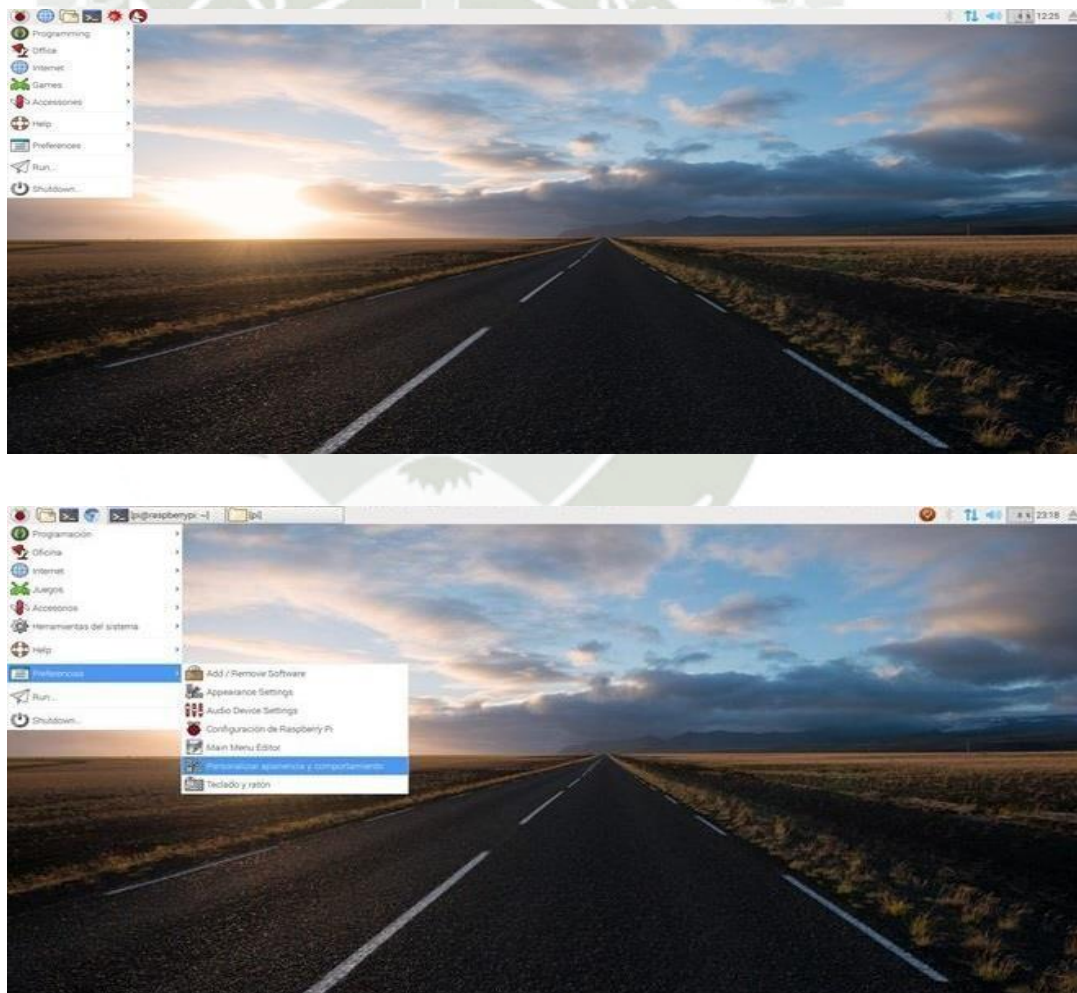
Raspbian (2012) afirma: "que fue desarrollado como un sistema operativo de código abierto, y está basado en Debian" (párr. 1). Este sistema ha sido optimizado para ser utilizado en la arquitectura ARM, que es la que maneja el

Raspberry Pi, proporcionando así un sistema operativo con un entorno amigable de fácil uso e instalación en el Raspberry Pi.

Raspbian (2012) afirma: “El sistema operativo tiene un entorno de escritorio LXDE, incluye herramientas de desarrollo muy interesantes como: IDLE para Python, Scratch, Wolfram Mathematica, Libre Oficce y un sin número de herramientas que podrás encontrar en su Pi Store” (párr. 2).

Figura 35

Escritorio Raspbian Jessie



Nota. Obtenido de: <https://www.raspberrypi.com/software/>

3.5 DESARROLLO EXPERIMENTAL

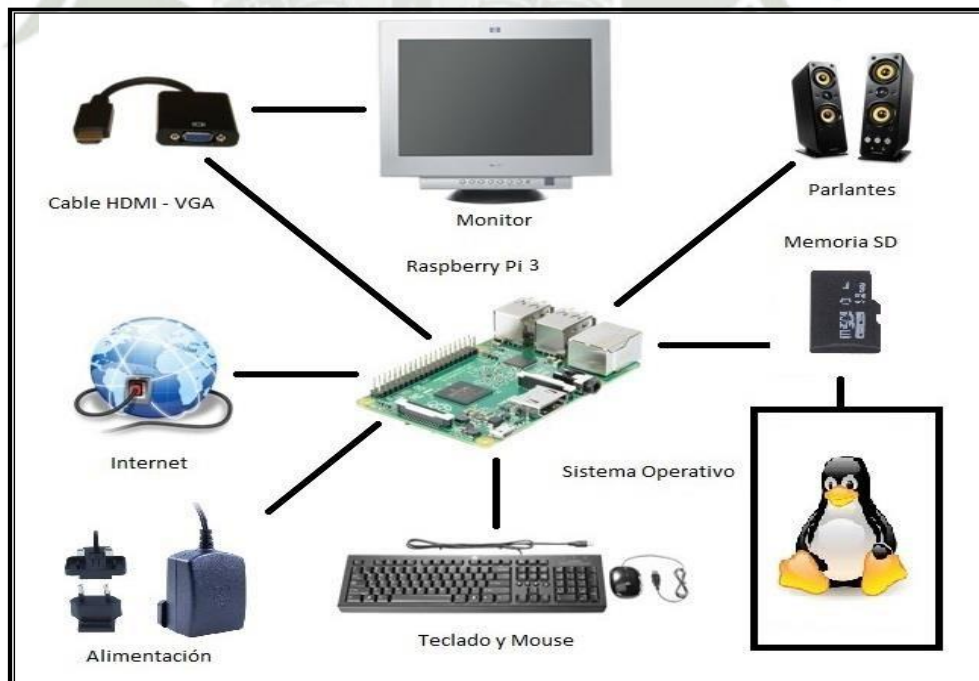
3.5

3.5.1 Diseño de proceso de interconexión entre los diferentes componentes

Una vez que se tuvo selecto el sistema operativo y los demás componentes a utilizar, procedemos a realizar la instalación en la placa Raspberry. Es necesario llevar a cabo la interconexión entre los componentes. En esta fase del proyecto, se elabora un esquema que ilustra la disposición y conexiones entre cada una de las partes del prototipo. Se especifica el uso de adaptadores para vincular los distintos componentes, creando un diagrama físico de interconexión que servirá de guía durante el montaje del prototipo.

Figura 36

Diseño De Interconexión De Componentes



Nota. Obtenido de: Varias Fuentes

3.5.2 Conexión de alimentación de Raspberry Pi

El prototipo representa un ordenador personal económico, motivo por el cual debe operar con una única fuente de energía que puede localizarse dentro del propio monitor. Para conectar la alimentación de la Raspberry Pi, proceda con la instalación. Una vez que el monitor esté cerrado, puede avanzar a insertar la memoria SD en la ranura designada para ello.

Figura 37

Inserción De Memoria SD En Raspberry Pi



Nota. Investigación Directa

Conecte el puerto VGA del monitor, esto puede presenta un considerable inconveniente, pues Raspberry Pi posee una entrada HDMI. Por ende, será necesario utilizar un convertidor VGA/HDMI. Gracias a este dispositivo, podemos transformar la señal analógica del monitor a digital.

Figura 38

Conversor VGA/HDMI o HDMI/HDMI

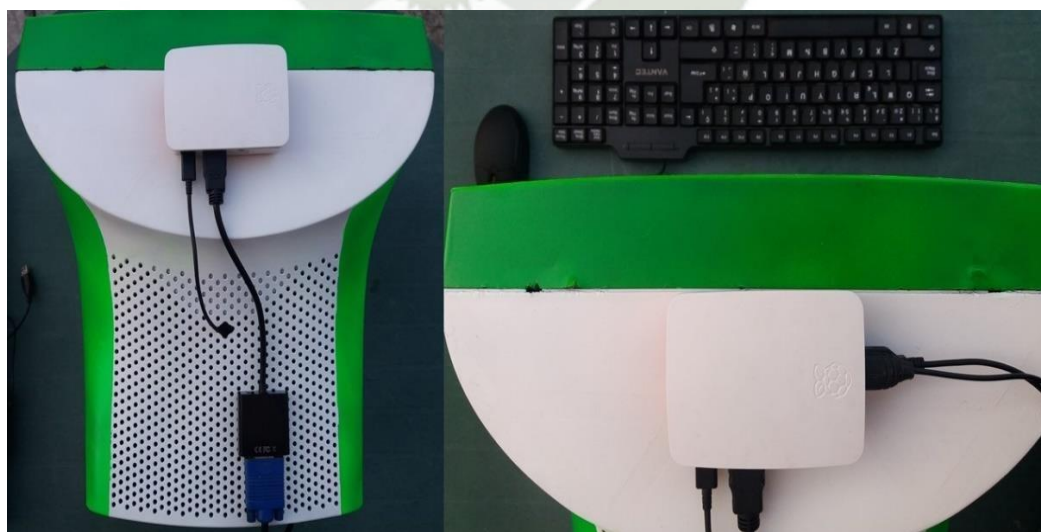


Nota. Investigación Directa

Ahora coloque la placa en la posición deseada, en este caso, en la parte superior del monitor. Conecte de forma adecuada la alimentación, la entrada de video, mouse y teclado a la Raspberry Pi para proceder con su encendido.

Figura 39

Acoplamiento Y Conexión De Raspberry Pi En Monitor



Nota. Investigación Directa

3.5.3 Encendido del prototipo

Después de haber conectado todos los componentes del prototipo del computador personal de bajo costo, proceda a enchufar la alimentación del monitor en el tomacorriente para encenderlo y permitir el arranque (boot) del sistema operativo de la Raspberry Pi.

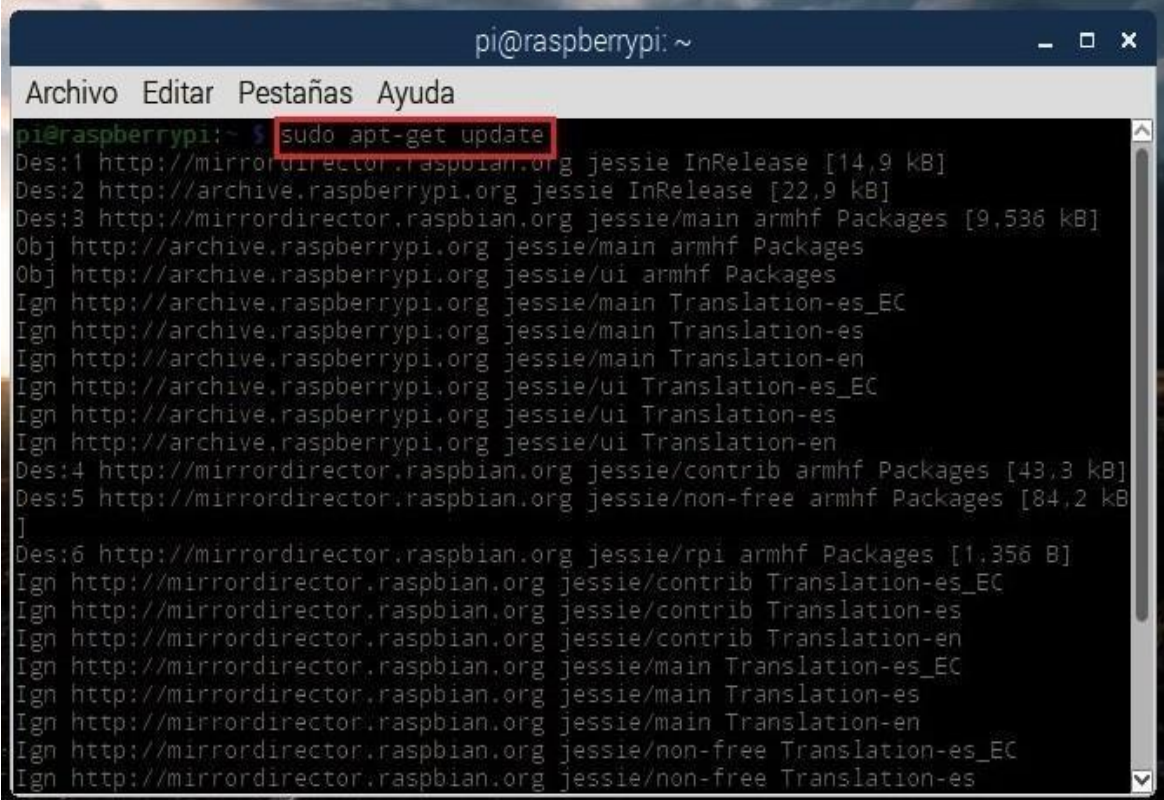
3.5.4 Configuración inicial del sistema operativo

Cada sistema operativo requiere configuración en su primer uso para un funcionamiento correcto. Por este motivo, a continuación, se presenta la configuración correspondiente de los sistemas operativos seleccionados para el prototipo. Es relevante mencionar que las configuraciones en versiones previas de estos sistemas solían ser más complicadas debido a la cantidad de comandos que requerían ejecutarse.

3.5.5 Configuración inicial de Raspbian

Raspbian, gracias a sus recientes actualizaciones en el sistema operativo, ahora se configura automáticamente para su uso, eliminando la necesidad de seguir varios pasos de configuración en versiones anteriores.

Una vez que el sistema operativo ha arrancado, es necesario acceder a la terminal y actualizar la caché de apt mediante el uso del comando `sudo apt-get update`, el cual actualiza la herramienta apt.

Figura 40*Actualización Cache De Apt Raspbian*

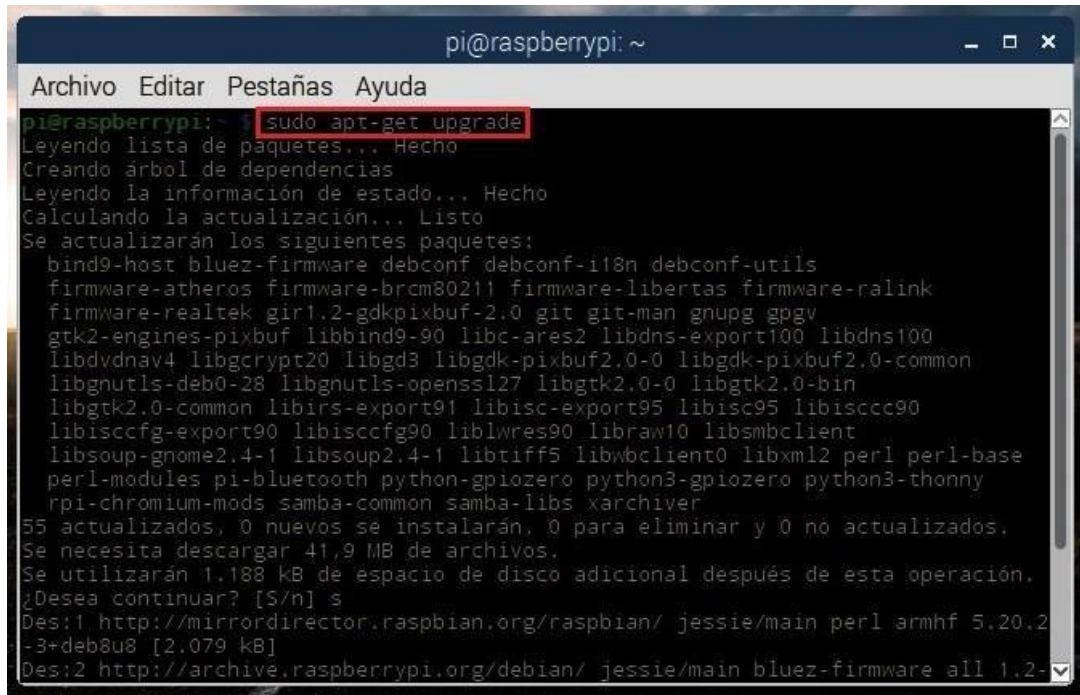
```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~$ sudo apt-get update  
Des:1 http://mirrordirector.raspbian.org jessie InRelease [14,9 kB]  
Des:2 http://archive.raspberrypi.org jessie InRelease [22,9 kB]  
Des:3 http://mirrordirector.raspbian.org jessie/main armhf Packages [9,536 kB]  
Obj http://archive.raspberrypi.org jessie/main armhf Packages  
Obj http://archive.raspberrypi.org jessie/ui armhf Packages  
Ign http://archive.raspberrypi.org jessie/main Translation-es_EC  
Ign http://archive.raspberrypi.org jessie/main Translation-es  
Ign http://archive.raspberrypi.org jessie/main Translation-en  
Ign http://archive.raspberrypi.org jessie/ui Translation-es_EC  
Ign http://archive.raspberrypi.org jessie/ui Translation-es  
Des:4 http://mirrordirector.raspbian.org jessie/contrib armhf Packages [43,3 kB]  
Des:5 http://mirrordirector.raspbian.org jessie/non-free armhf Packages [84,2 kB]  
]  
Des:6 http://mirrordirector.raspbian.org jessie/rpi armhf Packages [1,356 B]  
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-es_EC  
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-es  
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en  
Ign http://mirrordirector.raspbian.org jessie/main Translation-es_EC  
Ign http://mirrordirector.raspbian.org jessie/main Translation-es  
Ign http://mirrordirector.raspbian.org jessie/main Translation-en  
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-es_EC  
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-es
```

Nota. Investigación Directa

Después de actualizar la caché de apt, proceda a la actualización de todo el software de Raspberry Pi con el comando `sudo apt-get upgrade`. Este comando examina todos los paquetes instalados y proporciona detalles sobre las actualizaciones que deben realizarse en los paquetes.

Figura 41

Actualización De Todo El Software De Raspbian



```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~$ sudo apt-get upgrade  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Calculando la actualización... Listo  
Se actualizarán los siguientes paquetes:  
bind9-host bluez-firmware debconf debconf-i18n debconf-utils  
firmware-atheros firmware-brcm80211 firmware-libertas firmware-ralink  
firmware-realtek gir1.2-gdkpixbuf-2.0 git git-man gnupg gpgv  
gtk2-engines-pixbuf libbind9-90 libc-ares2 libdns-export100 libdns100  
libdvnav4 libgcrypt20 libgd3 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-common  
libgnutls-deb0-28 libgnutls-openssl127 libgtk2.0-0 libgtk2.0-bin  
libgtk2.0-common libirs-export91 libisc-export95 libisc95 libisccc90  
libisccfg-export90 libisccfg90 liblwres90 libraw10 libsmclient  
libsoup-gnome2.4-1 libsoup2.4-1 libtiff5 libwbclient0 libxml2 perl perl-base  
perl-modules pi-bluetooth python-gpiozero python3-gpiozero python3-thonny  
rpi-chromium-mods samba-common samba-libs xarchiver  
55 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.  
Se necesita descargar 41,9 MB de archivos.  
Se utilizarán 1.188 kB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n] s  
Des:1 http://mirrordirector.raspbian.org/raspbian/ jessie/main perl armhf 5.20.2  
-3+deb8u8 [2.079 kB]  
Des:2 http://archive.raspberrypi.org/debian/ jessie/main bluez-firmware all 1.2-
```

Nota. Investigación Directa

Una vez completado este procedimiento, el sistema estará listo para su uso y para la instalación de programas adicionales

3.5.6 THONNY PHYTON IDE

Thonny es un novedoso Entorno de Desarrollo Integrado (IDE) de Python creado para simplificar el proceso de aprendizaje y enseñanza de la programación, incorporando de manera inherente la visualización de programas en el flujo de trabajo de los principiantes. Entre sus características más destacadas se encuentran diferentes métodos para examinar el código, la evaluación gradual de expresiones, una representación visual clara de la pila de llamadas y un modo específicamente diseñado para explicar los conceptos de referencias y montón. Thonny es de uso gratuito y puede ser ampliado mediante

la incorporación de complementos. A continuación, vamos a ahondar de manera más específica en las particularidades que nos brinda Thonny.

3.5.6.1 Características del IDE Thonny

- Inicio sencillo. Thonny cuenta con Python 3.7 integrado, simplificando su instalación para aquellos que desean aprender a programar. También es compatible con instalaciones separadas de Python si se requieren. Su interfaz inicial está diseñada para evitar distracciones para los principiantes.
- Variables sin complicaciones. Después de los hello-worlds, se puede acceder a la opción Ver → Variables para observar cómo los comandos afectan las variables de Python.
- Depurador sencillo. Mediante combinaciones como Ctrl + F5, se puede ejecutar el programa paso a paso sin la necesidad de puntos de interrupción. Las combinaciones F6 y F7 permiten grandes y pequeños pasos respectivamente, siguiendo la estructura del programa, no solo las líneas de código.
- Evaluación de expresiones paso a paso. Utilizando pequeños pasos, permite observar cómo Python evalúa las expresiones mediante pasos pequeños, proporcionando una comprensión más clara del proceso.
- Representación fidedigna de llamadas a funciones. Al ingresar en una llamada de función, se muestra una ventana separada con una tabla de variables locales y un puntero de código independientes, facilitando la comprensión de las llamadas a funciones, especialmente en casos de recursividad.

- Detección de errores de sintaxis: Thonny destaca los errores comunes de sintaxis, como comillas sin cerrar o paréntesis, lo cual es útil para principiantes que están aprendiendo.
- Explicación de alcances. Se resaltan las variables para recordar que un mismo nombre no siempre representa la misma variable, ayudando a identificar errores tipográficos. Además, las variables locales se caracterizan por ser visualmente diferentes.
- Modo para explicar referencias. Las variables se presentan inicialmente según el modelo simplificado (nombre → valor), pero puede cambiar a un modelo más realista (nombre → dirección / id → valor).
- Completado de código. Los estudiantes pueden explorar las API con la ayuda de la función de completado de código.
- Sistema amigable para principiantes. Thonny simplifica la instalación de paquetes adicionales y el manejo de Python en la línea de comandos a través de la opción Herramientas → Abrir shell del sistema, encargándose de PATH y posibles conflictos con otros intérpretes de Python..
- Interfaz gráfica de usuario (GUI) PIP simple y limpia. La opción Herramientas → Administrar paquetes facilita aún más la instalación de paquetes de terceros de manera sencilla y limpia.

Figura 42*Thonny Phyton.*

Nota. Obtenido de: (Corredera, 2023)

Este IDE, nos permitirá realizar la programación y en él podríamos experimentar hasta poder entrenar de forma eficiente nuestra Raspberry.

3.5.7 PROGRAMACION Y EJECUCION

Se tiene previsto realizar un programa con las características mínimas que se propuso en el plan estratégico de desarrollo.

Consta de cuatro elementos principales para la detección:

- Reconocimiento de Mascarilla
- Reconocimiento de Lentes
- Reconocimiento de Casco
- Reconocimiento de Casco y Lentes

De los cuales se tiene considerado como una mejora, la integración de dos elementos en una sola imagen para su procesamiento y detección.

- Al inicio se tiene el comando “import cv2” que viene a ser una librería de visión por computador de OpenCV, biblioteca de visión artificial de código abierto, inicialmente creada por Intel. Su nombre deriva de Open Computer Vision (Visión Artificial Abierta). Sus aplicaciones incluyen el reconocimiento de objetos, la detección de movimiento, y la generación de reconstrucciones en 3D a partir de imágenes.
- “import os” que también es una librería para acceder a funciones del sistema operativo. El módulo "os" de Python permite ejecutar operaciones dependientes del sistema operativo, tales como la creación de carpetas, , obtener información sobre un proceso, visualización de los contenidos de un directorio y finalizar un proceso, entre otras funciones
- NumPy, utilizado para operaciones matemáticas y de matriz.
- ‘Imutils’: Una biblioteca que simplifica algunas operaciones comunes de OpenCV.
- Se definen las coordenadas (x1, y1) y (x2, y2) que forman un rectángulo en el video. Este rectángulo se utilizará para seleccionar un objeto de interés.
- Comienza un bucle infinito (while True) para procesar los fotogramas del video.
- Se lee un fotograma desde la cámara web con ‘cap.read()’. La variable ‘ret’ indica si la lectura fue exitosa y ‘frame’ contiene el fotograma actual.
- Se voltea el fotograma horizontalmente con cv2.flip(frame, 1) para que la vista en pantalla sea como un espejo.

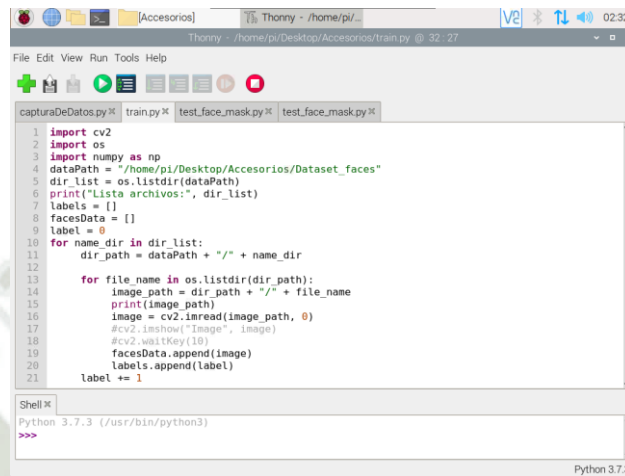
- Se dibuja un rectángulo azul alrededor del objeto de interés en el fotograma original usando `'cv2.rectangle'`.
- Se redimensiona el objeto recortado a un ancho de 72 píxeles utilizando `'imutils.resize'`.
- El código espera una tecla presionada con `'k = cv2.waitKey(1)'`.
- Si la tecla presionada es 's', se guarda la imagen del objeto recortado en la carpeta "Datos" con un nombre de archivo que incluye el contador `'count'`. Luego se incrementa el contador.
- Se muestra el fotograma original con el rectángulo y el objeto recortado en dos ventanas separadas utilizando `'cv2.imshow'`.
- Finalmente, después de salir del bucle, se libera la captura de video con `'cap.release()'` y se cierran todas las ventanas de visualización con `'cv2.destroyAllWindows()'`.

3.5.7.2 Entrenamiento del modelo en un archivo XML

Entrenamos el modelo con un detector de rostros utilizando el algoritmo LBPH (Local Binary Pattern Histograms) en OpenCV y luego guardamos el modelo entrenado en un archivo XML.

Figura 44

Entorno de Programación para entrenar modelos.



```

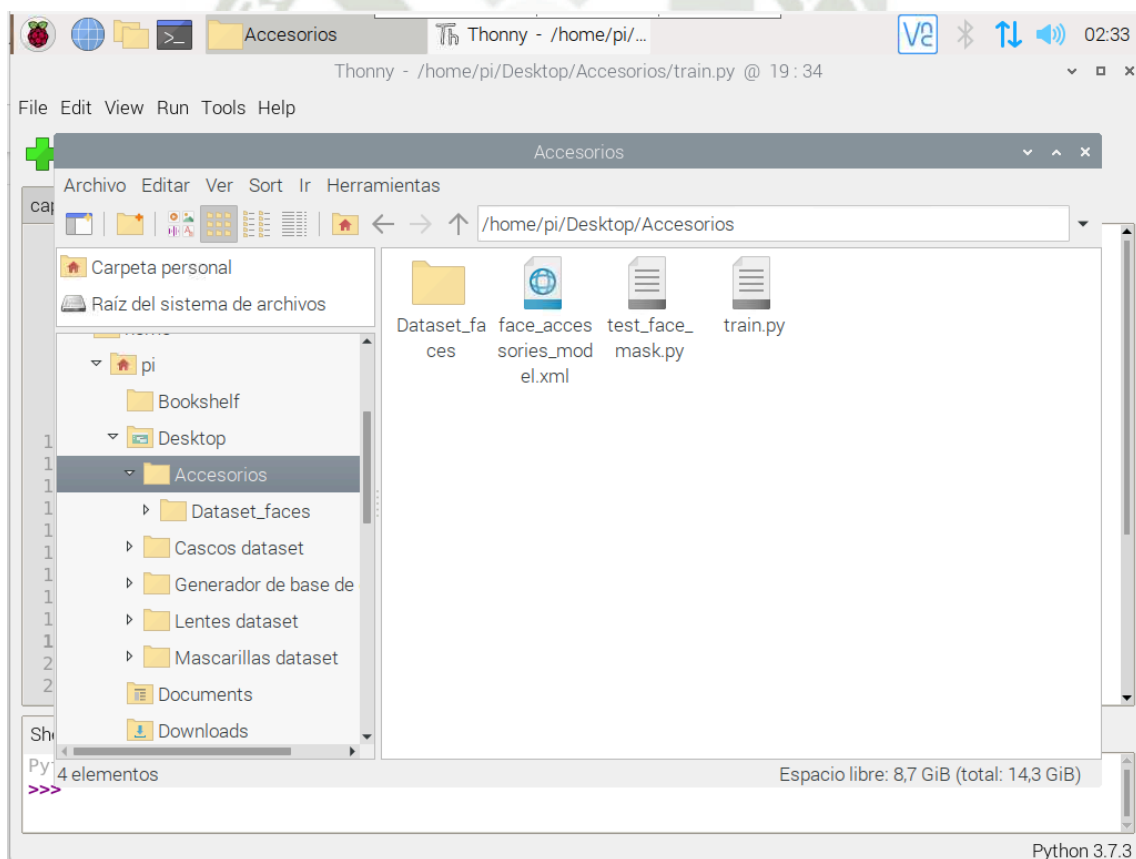
1 import cv2
2 import os
3 import numpy as np
4 dataPath = "/home/pi/Desktop/Accesorios/Dataset_faces"
5 dir_list = os.listdir(dataPath)
6 print("Lista archivos:", dir_list)
7 labels = []
8 facesData = []
9 label = 0
10 for name_dir in dir_list:
11     dir_path = dataPath + "/" + name_dir
12
13     for file_name in os.listdir(dir_path):
14         image_path = dir_path + "/" + file_name
15         print(image_path)
16         image = cv2.imread(image_path, 0)
17         #cv2.imshow("Image", image)
18         #cv2.waitKey(10)
19         facesData.append(image)
20         labels.append(label)
21     label += 1
    
```

- Primero se importan las mismas librerías de OpenCV, os y NumPy.
- Se define la variable “dataPath” que contiene la ruta a la carpeta donde se encuentran los fotogramas utilizados para entrenar el reconocedor de rostros.
- Se obtiene una lista de nombres de archivos y directorios en “dataPath” utilizando “os.listdir(dataPath)”. Estos nombres de directorio representarán las etiquetas de las imágenes.
- Se realiza un bucle sobre cada directorio en “dir_list” (que son los nombres de los subdirectorios en “dataPath”).
- Se imprime el número de imágenes asociadas a cada etiqueta utilizando la función “np.count_nonzero” en combinación con “np.array(labels)” y se muestra cuántas imágenes hay para cada etiqueta.

- Se crea un objeto declarado como “face_mask” y haciendo uso de “cv2.face.LBPHFaceRecognizer_create()”. Este objeto se usará para entrenar y reconocer rostros utilizando el algoritmo LBPH.
- Se entrena el modelo de reconocimiento de rostros utilizando el método “train” del objeto “face_mask”, proporcionando “facesData” (las imágenes) y “np.array(labels)” (las etiquetas).

Figura 45

Archivo XML generado por el modelo entrenado

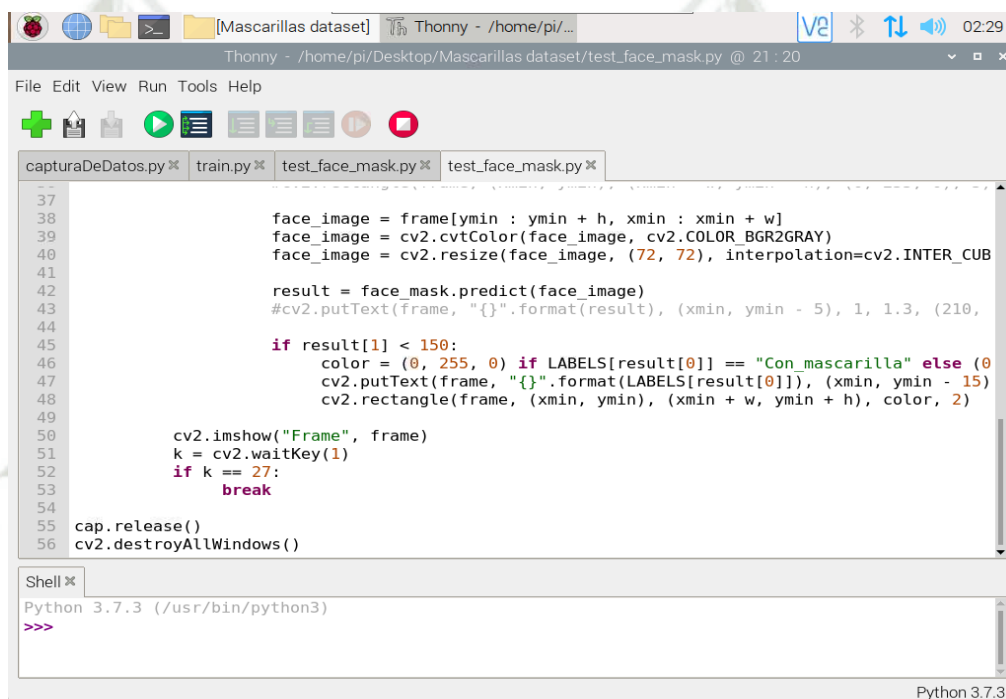


3.5.7.3 Sistema de detección de EPP

Para realizar las predicciones usaremos el modelo previamente entrenado para detectar rostros en un video en tiempo real, clasificarlos en diferentes categorías (etiquetas) y mostrar el resultado en el video.

Figura 46

Entorno de Programación Detección de Mascarilla

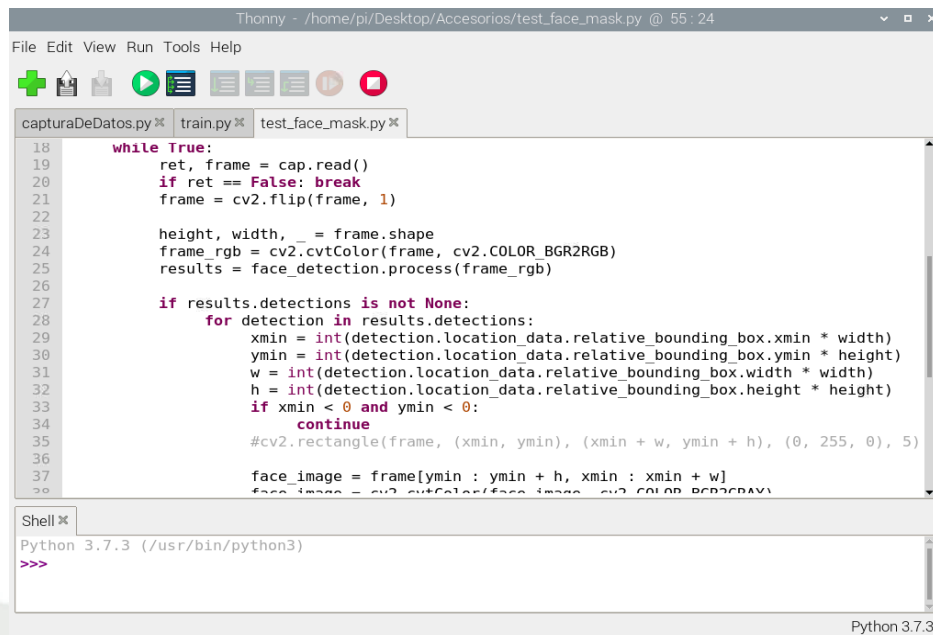


```
Thonny - /home/pi/Desktop/Mascarillas dataset/test_face_mask.py @ 21 : 20
Thonny - /home/pi/Desktop/Mascarillas dataset/test_face_mask.py @ 21 : 20
File Edit View Run Tools Help
+ [Icons]
capturaDeDatos.py x train.py x test_face_mask.py x test_face_mask.py x
37
38     face_image = frame[ymin : ymin + h, xmin : xmin + w]
39     face_image = cv2.cvtColor(face_image, cv2.COLOR_BGR2GRAY)
40     face_image = cv2.resize(face_image, (72, 72), interpolation=cv2.INTER_CUB
41
42     result = face_mask.predict(face_image)
43     #cv2.putText(frame, "{}".format(result), (xmin, ymin - 5), 1, 1.3, (210,
44
45     if result[1] < 150:
46         color = (0, 255, 0) if LABELS[result[0]] == "Con_mascarilla" else (0
47         cv2.putText(frame, "{}".format(LABELS[result[0]]), (xmin, ymin - 15)
48         cv2.rectangle(frame, (xmin, ymin), (xmin + w, ymin + h), color, 2)
49
50     cv2.imshow("Frame", frame)
51     k = cv2.waitKey(1)
52     if k == 27:
53         break
54
55     cap.release()
56     cv2.destroyAllWindows()

Shell x
Python 3.7.3 (/usr/bin/python3)
>>>
```

Figura 47

Entorno de Programación Detección Múltiple



```

Thonny - /home/pi/Desktop/Accesorios/test_face_mask.py @ 55:24
File Edit View Run Tools Help
+ [Icons]
capturaDeDatos.py train.py test_face_mask.py
18     while True:
19         ret, frame = cap.read()
20         if ret == False: break
21         frame = cv2.flip(frame, 1)
22
23         height, width, _ = frame.shape
24         frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
25         results = face_detection.process(frame_rgb)
26
27         if results.detections is not None:
28             for detection in results.detections:
29                 xmin = int(detection.location_data.relative_bounding_box.xmin * width)
30                 ymin = int(detection.location_data.relative_bounding_box.ymin * height)
31                 w = int(detection.location_data.relative_bounding_box.width * width)
32                 h = int(detection.location_data.relative_bounding_box.height * height)
33                 if xmin < 0 and ymin < 0:
34                     continue
35                 #cv2.rectangle(frame, (xmin, ymin), (xmin + w, ymin + h), (0, 255, 0), 5)
36
37                 face_image = frame[ymin : ymin + h, xmin : xmin + w]
38                 face_image = cv2.cvtColor(face_image, cv2.COLOR_BGR2GRAY)
39
Shell
Python 3.7.3 (/usr/bin/python3)
>>>
Python 3.7.3

```

Para la explicación del código principal (Sistema de detección de EPP) se detallará las líneas importantes para un mejor entendimiento.

- Primero importamos las librerías ya mencionadas como “cv2” y “os”.
- Luego tenemos el “import mediapipe as mp” que llama a la librería de mediapipe para reconocimiento facial.
- mp_face_detection=mp.solutions.face_detection #Se asigna una variable para la detección de rostros mediante mediapipe.
- LABELS=["sin_accesorios", "con lentes", "con casco"] #Ser crea labels para asignar valores de "0", "1", "2" según el orden del vector.
- #Leer el modelo
- face_mask=cv2.face.LBPHFaceRecognizer_create()
- face_mask.read("face_accesories_model.xml") #Lee el modelo previamente entrenado para a visión artificial.

- `cap=cv2.VideoCapture(0) #Lee los frames de la web-cam (captura de video de la web-cam) en tiempo real`
- `with mp_face_detection.FaceDetection(`
- `min_detection_confidence=0.5) as face_detection: #Se le asigna 0.5 como mínimo de fidelidad de reconocimiento fácil (50%)`
- `while True:`
- `ret,frame=cap.read() #leemos los frames del video capturado en tiempo real y guardamos los valores en las variables ret y frame: ret : si la lectura es correcta o no, frame: la imagen (foto/frame) tomada en el momento`
- `if ret==False:break #Si la lectura no es correcta salimos del while`
- `frame=cv2.flip(frame,1) #Voltea la imagen para que se vea como si nos viéramos en un espejo`
- `height, width, _= frame.shape # se lee el alto y ancho de cada imagen capturada del video en tieif results.detections is not None: #Si el resultado es diferente de vacio entonces:`
- `for detection in results.detections: #Para los resultados que si se encuentren en las detecciones`
- `xmin=int(detection.location_data.relative_bounding_box.xmin*width) #Generamos los puntos para dibujar el cuadrado que bordea al rostro - primer punto`
- `ymin=int(detection.location_data.relative_bounding_box.ymin*height) #Generamos los puntos para dibujar el cuadrado que bordea al rostro - segundo punto`

- `w=int(detection.location_data.relative_bounding_box.width*width)`
#Generamos los puntos para dibujar el cuadrado que bordea al rostro - tercer punto
- `h=int(detection.location_data.relative_bounding_box.height*height)`
#Generamos los puntos para dibujar el cuadrado que bordea al rostro - cuarto punto
- `if xmin<0 and ymin<0: #si llegan a aparecer valores de dimensión negativas, menores a cero nos saltamos estas imágenes porque son errores`
- `face_image=frame[ymin:ymin+h,xmin:xmin+w] #Dibujamos el rectángulo en la imagen en base a los puntos generados previamente`
- `face_image=cv2.cvtColor(face_image,cv2.COLOR_BGR2GRAY)`
#Convertimos la imagen a escala de grises
- `face_image=cv2.resize(face.image, (72,72), interpolation=cv2.INTER_CUBIC)` #Cortamos la imagen leída a un tamaño pequeño de 72 x 72mm ya que debe ser del mismo tamaño con el que se entrenó el modelo
- `result=face_mask.predict(face.image) #leemos la predicción del modelo`
- `#cv2.putText(frame, "{}".format(result), (xmin,ymin-5),1,1.3(210,124,176),cv2.LINE_AA)`
- `if result[1]<150: # 150 es el primer valor de fidelidad`
- `color=(0,0,255) if LABELS[result[0]]=="Sin accesorios" else (0,255,0)`
#Pintamos el recuadro de color verde o rojo según corresponda

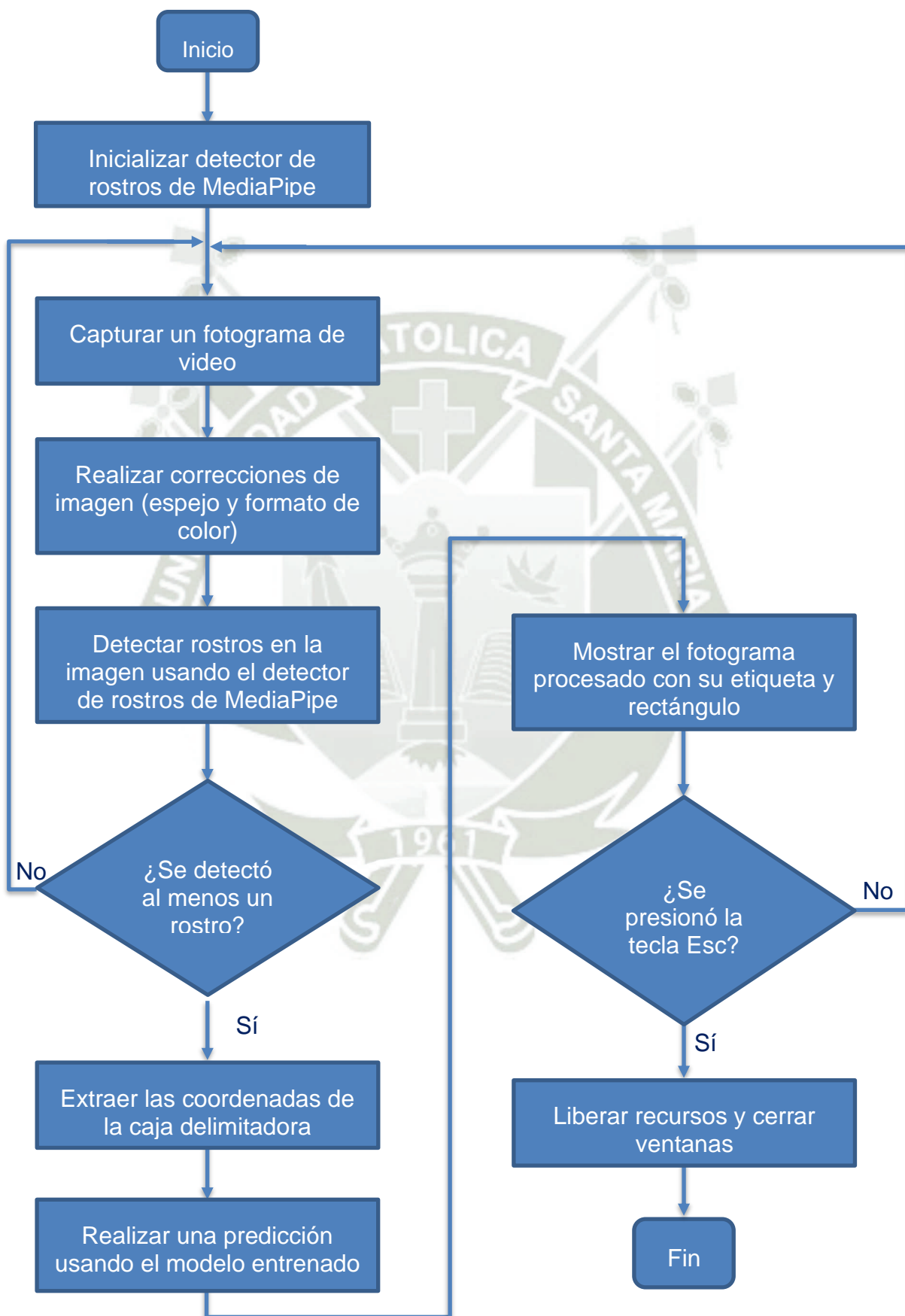
- `cv2.putText(frame,"{}".format(LABELS[result[0]]), (xmin,ymin-15),2,1,color,1,cv2.LINE_AA) # escribimos los label (con casco, con lentes, sin accesorios, etc) según a la predicción del modelo`
- `cv2.rectangle(frame, (xmin,ymin),(xmin+w,ymin+h),color,2) #Generamos la imagen ya con el cuadro y el color generado previamente`
- `cv2.imshow("frame", frame) #imprimimos la imagen, con cuadro, label y color`
- `k=cv2.waitKey(1) #Espera a que se presiona alguna tecla`

A continuación, se tiene el diagrama de flujo del Reconocimiento en tiempo real:



Figura 48

Diagrama de flujo del reconocimiento en tiempo real

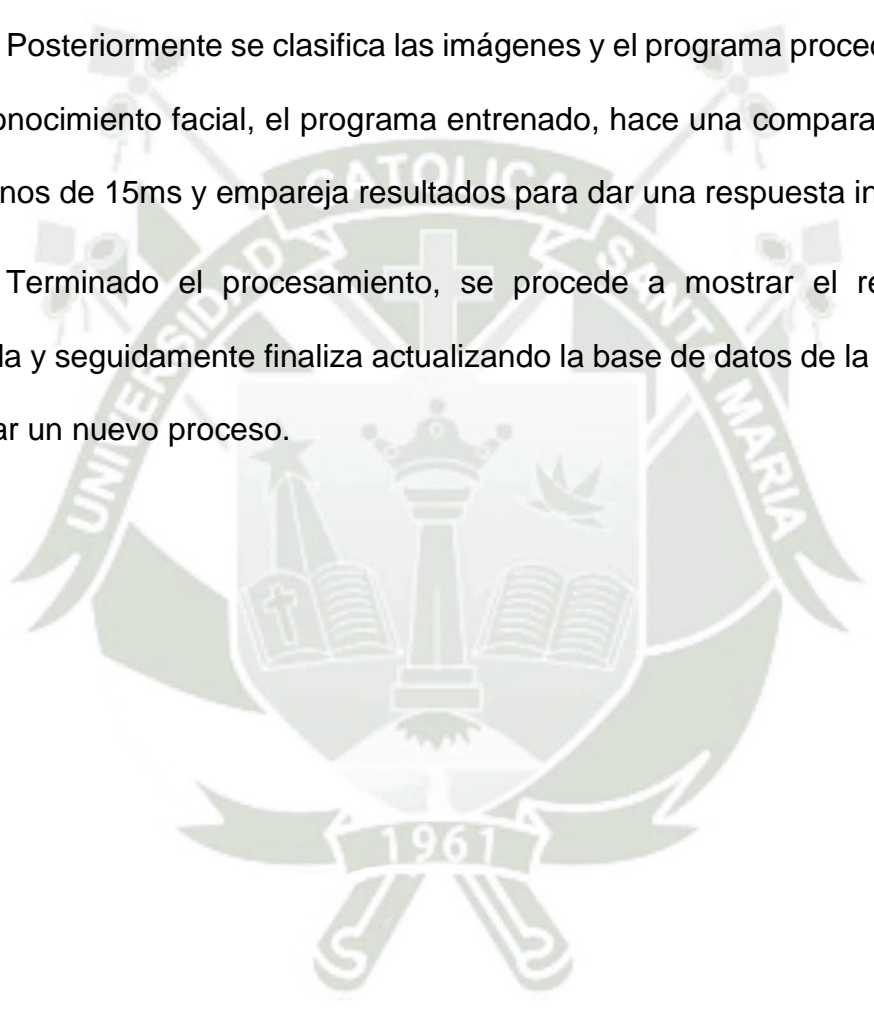


Nota. Elaboración Propia

Al iniciar el programa en la Rapsberry, se procede a capturar la imagen, internamente se detecta el objeto y se procede a extraer cada característica que anteriormente fue entrenada en la base de datos.

Posteriormente se clasifica las imágenes y el programa procede a realizar el reconocimiento facial, el programa entrenado, hace una comparación interna en menos de 15ms y empareja resultados para dar una respuesta inmediata.

Terminado el procesamiento, se procede a mostrar el resultado en pantalla y seguidamente finaliza actualizando la base de datos de la misma para esperar un nuevo proceso.



CAPITULO IV: RESULTADOS

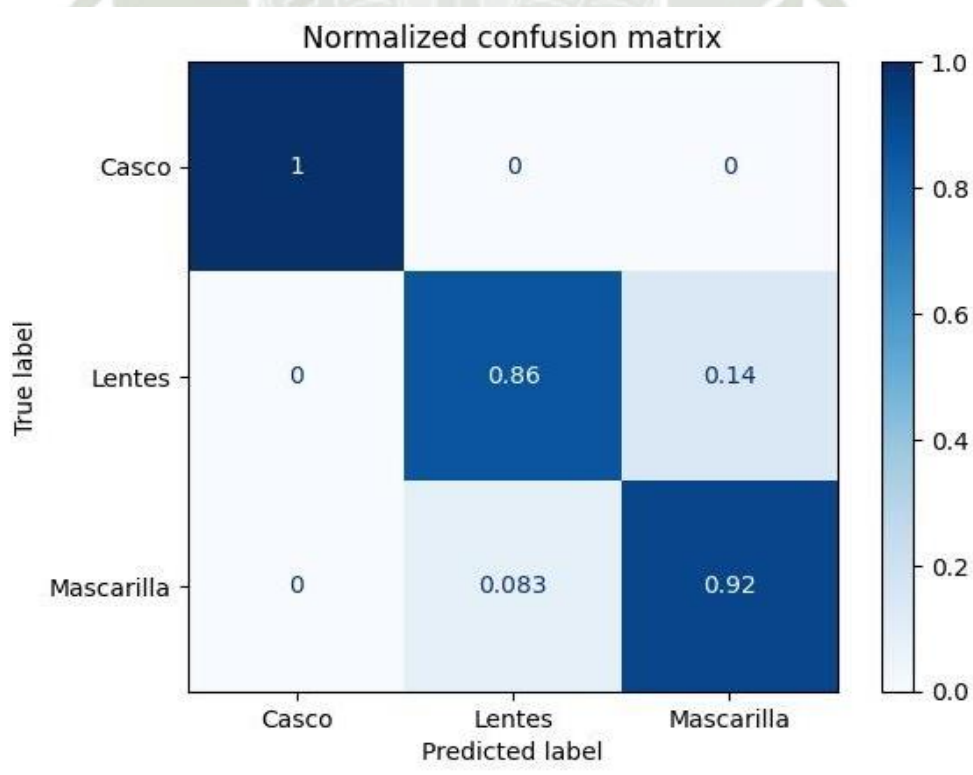
En base de la Figura 50 hasta la

Figura 54 , se tiene una detección con alto índice de fiabilidad para la mascarilla, lentes y casco.

Para conocer el porcentaje de fiabilidad hacemos uso de la matriz de confusión. Esta matriz evalúa la precisión de la clasificación, en la diagonal se muestra la tasa de clasificaciones o predicciones hechas correctamente (Verdaderos positivos, verdaderos negativos). Los falsos positivos y falsos negativos se encuentran en las otras posiciones de la matriz de confusión (Clasificaciones incorrectas).

Figura 49

Matriz de confusión



La Figura 49 muestra que la clase casco tiene una precisión del 100%, la clase lentes tiene una precisión del 86% y la clase mascarilla tiene una precisión del 92%.

Figura 50

Entorno de Programación “Detección de Mascarilla”

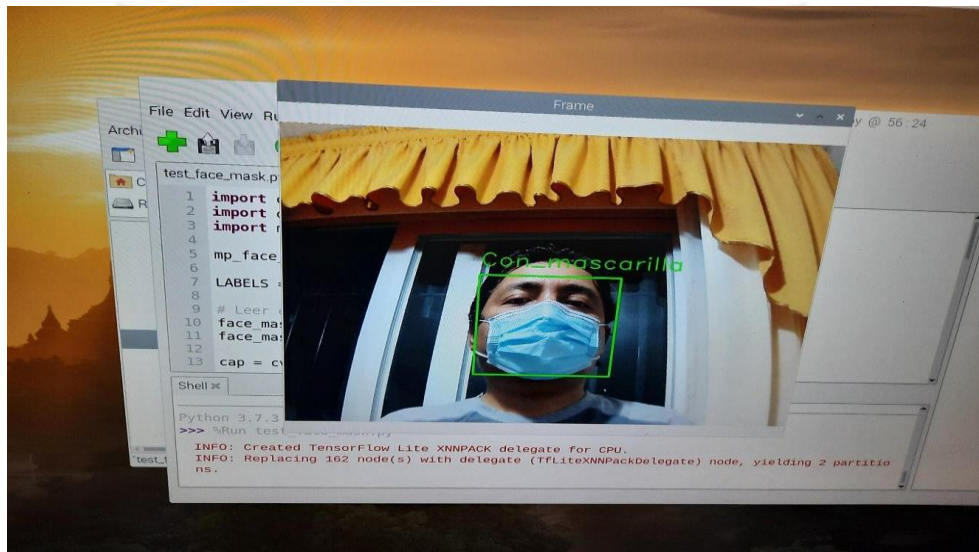


Figura 51

Entorno de Programación “Detección de Lentes”

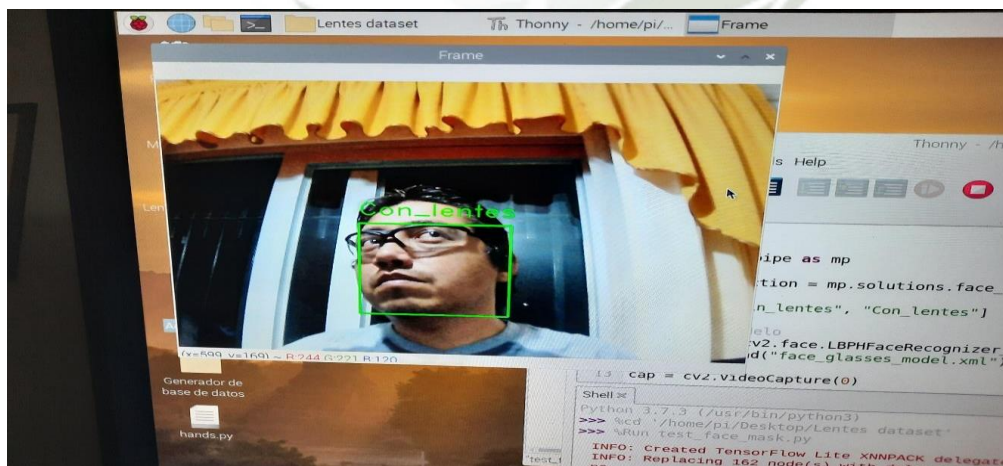


Figura 52

Entorno de Programación “Detección de Casco”

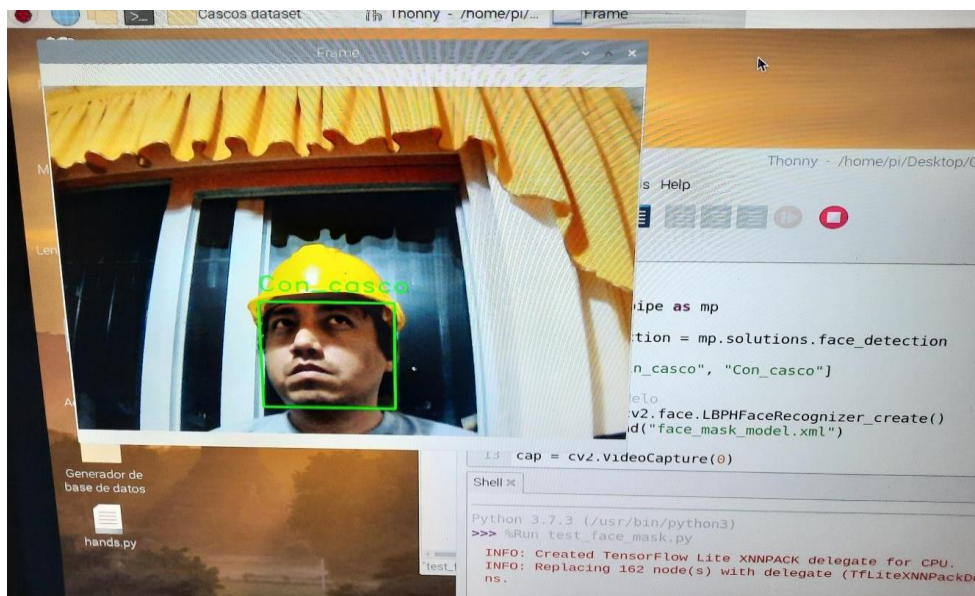


Figura 53

Entorno de Programación "Detección de Casco"

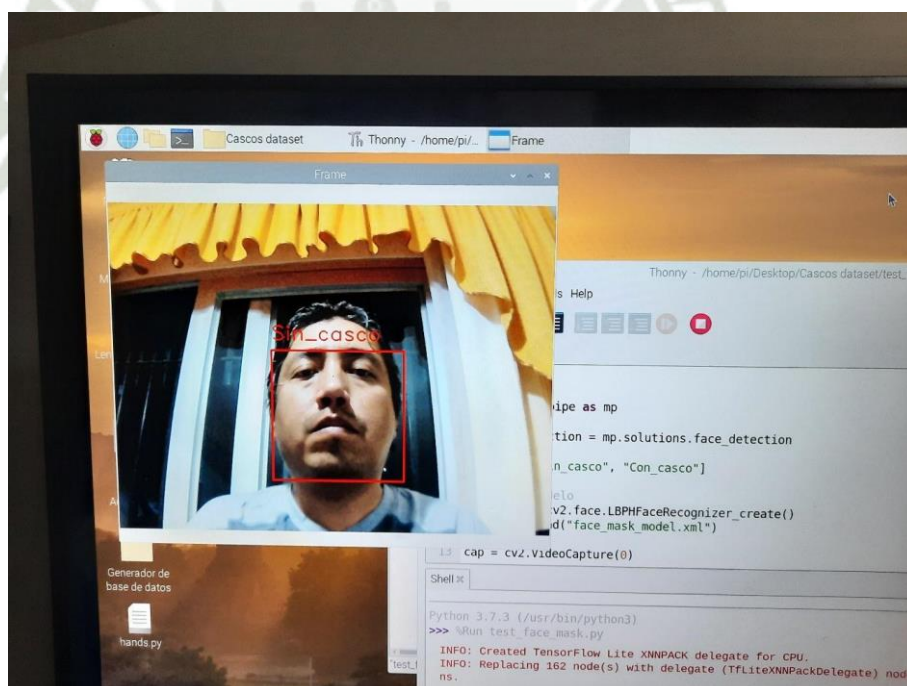
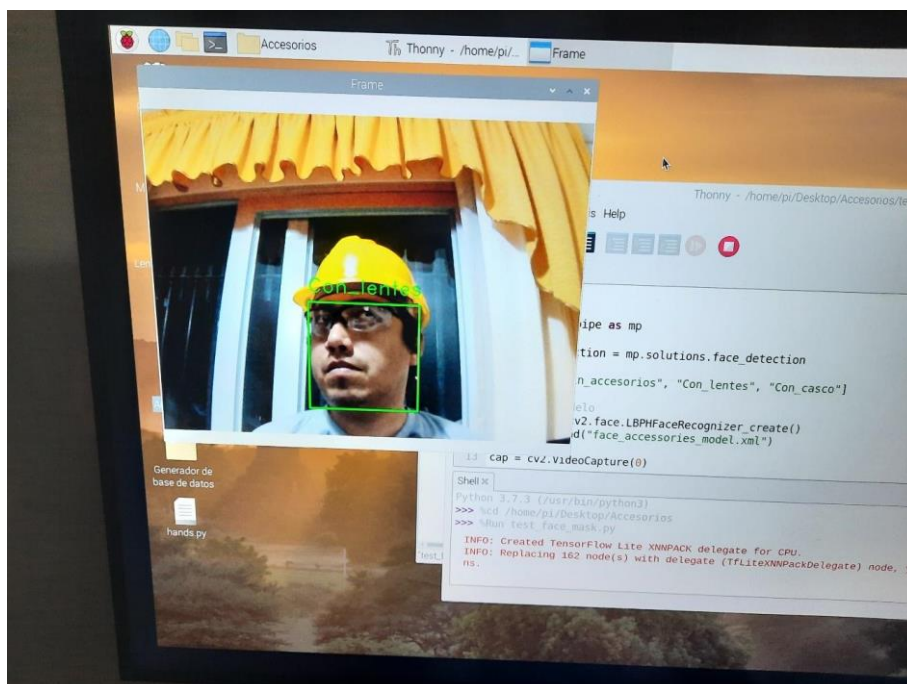


Figura 54

Entorno de Programación "Detección de Lentes"



En la sección 3.5.7 Programación y ejecución se indica que el proceso de detección y clasificación de imágenes es de alrededor de 15 ms con estos resultados, podemos afirmar que la clasificación de EPP es en tiempo real en vista a la rapidez del proceso.

Para el entrenamiento del clasificador LBPH, se utilizó imágenes de entrenamiento: para la clase caso 802, para la clase lentes 816, para la clase mascarilla 795.

Para el test se utilizó imágenes de prueba: para la clase caso 141, para la clase lentes 144, para la clase mascarilla 140. Donde estas imágenes representan el 15% del total

Tabla 7

Test

	CASCOS	LENTES	MASCARILLA
Imágenes de	802	816	795

entrenamiento			
Imágenes de prueba	141	144	140
TOTAL	943	960	935



CAPITULO V: PRESUPUESTO

El presupuesto creado está dividido en diversas secciones, que incluyen el componente de software y hardware, así como el apartado de gastos relacionados con el personal. Estas secciones conforman el costo global del proyecto.

5.1 Coste de Hardware.

En la siguiente tabla se muestra los costos de los materiales necesarios para implementar nuestro sistema de detección con visión artificial.

Tabla 8

Costos del hardware.

Id	Elemento / Descripción	Cantidad	P.U.	Valor
01	Raspberry Pi 3 Model B+ Con Kit Accesorios Completo Portátil empleado para el proyecto	1	372.34	372.34
02	Monitor Monitor HD de 19" con entrada HDMI	1	250.00	250.00
03	2K HD Camera Cámara webcam	1	85.90	85.90
04	Cable HDMI Conexión entre el raspberry y un monitor	1	10.00	10.00

5.2 Coste de Software

La totalidad del software empleado en el proyecto es de código abierto, lo que significa que es gratuito para todos los usuarios. Por lo tanto, no se asignará ningún presupuesto a esta categoría.

Tabla 9

Costos del software.

Id	Elemento / Descripción	Cantidad	P.U.	Valor
01	Raspbian Sistema Operativo	1	0	0

02	Thonny Entorno de programación	1	0	0
03	OpenCV Librerías de visión	1	0	0
04	OpenCV contrib Módulos extras de visión	1	0	0
05	MediaPipe Módulo aprendizaje automático multimodal	1	0	0

5.3 Coste de Personal

Para determinar el costo humano, se elaborará el presupuesto basándose en el salario que percibe típicamente un recién graduado en Ingeniería, utilizando una tarifa de 30 soles por hora para el sueldo de un ingeniero junior.

Tabla 10

Costos del personal

Id	Elemento / Descripción	Cantidad	P/h	Valor
01	Mano de Obra Instalaciones, implementación de los programas, pruebas de simulación y redacción de memoria.	80	30	2400

Para realizar la simulación y pruebas antes de pasar al campo se van a necesitar los siguientes materiales EPP para poner a prueba nuestro sistema.

Tabla 11

Costos de los EPP

Id	Elemento / Descripción	Cantidad	P/h	Valor
01	Casco	1	15.90	15.90
02	Lentes de seguridad 3m	1	12.90	12.90
03	Mascarilla	1	89.00	89.00

5.4 Gasto global

Se presenta el resumen del presupuesto para la implementación de un sistema de visión artificial para la detección de EPP.

Tabla 12*Resumen del presupuesto total.*

Descripción	Valor	Unidades
Costo de Hardware	718.24	S/.
Costo de Software	0.00	S/.
Costo de Mano de obra	2400.00	S/.
Costo de EEP	117.80	S/.
TOTAL	3236.04	S/.



CONCLUSIONES

1. En este proyecto se exploraron diversos métodos de visión artificial para la detección de equipos de protección personal (EPP). El algoritmo LBPH (Histogramas de Patrones Binarios Locales) fue seleccionado debido a su eficiencia y precisión en el reconocimiento de rostros y objetos, así como por su bajo costo computacional. Este método demostró una precisión del 92% en la detección de EPP bajo condiciones variadas de iluminación y fondos.
2. Los algoritmos se implementaron en Python y se ejecutaron en la plataforma Raspberry Pi 3B+, utilizando el sistema operativo Raspbian. Esta combinación permitió aprovechar las capacidades de procesamiento del Raspberry Pi, manteniendo un sistema económico y de fácil despliegue. Se desarrollaron scripts específicos para la captura, procesamiento y análisis de imágenes en tiempo real, demostrando la viabilidad del uso de hardware accesible y software de código abierto en aplicaciones de visión artificial.
3. Se eligieron las librerías OpenCV, Numpy y Mediapipe como herramientas clave para el procesamiento de imágenes. OpenCV ofreció una amplia gama de funciones para el análisis y detección de características visuales, mientras que Numpy facilitó la manipulación eficiente de datos matriciales. Mediapipe, con sus soluciones preentrenadas para la detección de rostros y manos, complementó el sistema al mejorar la precisión y robustez en la identificación de EPP.
4. Los algoritmos fueron implementados en Python y ejecutados en la plataforma Raspberry Pi 3B+, utilizando el sistema operativo Raspbian. Esta combinación permitió aprovechar las capacidades de procesamiento del Raspberry Pi, manteniendo un sistema económico y de fácil despliegue. Se

desarrollaron scripts específicos para la captura, procesamiento y análisis de imágenes en tiempo real, con un tiempo de procesamiento de aproximadamente 15 ms por imagen, demostrando la viabilidad del uso de hardware accesible y software de código abierto en aplicaciones de visión artificial.

5. Se llevaron a cabo pruebas exhaustivas en distintos ambientes para evaluar los algoritmos implementados. El algoritmo LBPH alcanzó una buena precisión de detección de EPP, destacando por su capacidad para operar eficazmente bajo variaciones de iluminación y oclusiones parciales de los equipos.
6. Se llevaron a cabo pruebas exhaustivas en distintos ambientes para evaluar los algoritmos implementados. El algoritmo LBPH alcanzó una precisión del 92% en la detección de EPP, destacando por su capacidad para operar eficazmente bajo variaciones de iluminación y oclusiones parciales de los equipos.
7. Se identificaron los tipos de EPP más críticos en el entorno industrial, como cascos, mascarillas y gafas de seguridad. Se creó una base de datos de imágenes de estos EPP, con 802 imágenes de cascos, 816 de lentes y 795 de mascarillas utilizadas para entrenar el modelo de detección. El procesamiento de estas imágenes incluyó técnicas de normalización, aumento de datos y segmentación, asegurando que el modelo fuera robusto frente a variaciones en el entorno y la apariencia de los EPP.
8. Se desarrolló un sistema completo de inspección de EPP que incluye la generación de una base de datos de entrenamiento, la creación de un modelo entrenado en formato XML y la implementación de un sistema de

detección en tiempo real. El sistema fue capaz de detectar con precisión el uso correcto de EPP y su diseño modular permite la futura integración de más tipos de EPP y mejoras funcionales. Este desarrollo no solo satisface las necesidades actuales de seguridad laboral, sino que también proporciona una plataforma escalable para aplicaciones futuras.

9. La matriz de confusión normalizada muestra que nuestro sistema de detección de equipos de protección personal (EPP) es altamente eficaz, logrando una precisión del 100% en la detección de cascos sin errores, un 86% en lentes de seguridad con un 14% de instancias incorrectamente clasificadas como mascarillas, y un 92% en mascarillas con un 8.3% de instancias clasificadas incorrectamente como lentes. Estos resultados valida la efectividad de los métodos y algoritmos implementados, subrayando la importancia de la detección precisa de EPP para la seguridad en el trabajo.
10. El sistema desarrollado demuestra una capacidad de clasificación en tiempo real, con un tiempo de procesamiento de aproximadamente 15 ms por imagen. Utilizando el clasificador LBPH, se entrenaron modelos con un conjunto de datos compuesto por 802 imágenes de cascos, 816 de lentes y 795 de mascarillas, mientras que para las pruebas se emplearon 141 imágenes de cascos, 144 de lentes y 140 de mascarillas, representando un 15% del total de imágenes disponibles. Estos resultados indican que el sistema es robusto y eficiente en condiciones reales, garantizando una detección precisa y rápida de equipos de protección personal.

RECOMENDACIONES

1. Es crucial llevar a cabo pruebas y validaciones en entornos industriales reales para garantizar la efectividad y fiabilidad del sistema de detección de EPP. Esto asegurará que el sistema cumpla con los estándares de seguridad requeridos y permita su implementación exitosa en diversos contextos industriales.
2. Mantener un formato de compresión adecuado para las imágenes almacenadas como muestras es fundamental para evitar la sobrecarga del espacio de memoria interna. Se sugiere utilizar formatos eficientes que balanceen calidad y tamaño de archivo.
3. Es recomendable realizar sesiones de entrenamiento y calibración del sistema directamente en la zona destinada para su uso final. Esto ayudará a ajustar y optimizar el rendimiento del sistema en condiciones específicas del entorno.
4. Para mejorar el rendimiento, se sugiere utilizar modelos más recientes de Raspberry Pi, como la Raspberry Pi 4, que ofrecen mayor potencia de procesamiento y capacidades mejoradas.
5. En entornos donde el acceso a la corriente eléctrica no es constante, se recomienda implementar soluciones para la gestión de energía, como el uso de baterías externas o sistemas de alimentación ininterrumpida (UPS).
6. Se recomienda considerar el uso de una Raspberry Pi con mayor cantidad de memoria RAM o explorar alternativas de hardware con mejor rendimiento. Además, evaluar la posibilidad de utilizar otro sistema operativo que pueda ofrecer mayor estabilidad y eficiencia.

7. Para mejorar la precisión del sistema de detección, se aconseja incrementar el número de muestras utilizadas en el entrenamiento del modelo. Una mayor cantidad de datos puede resultar en un sistema más robusto y preciso
8. Se recomienda monitorear la temperatura de la Raspberry Pi, especialmente durante la ejecución de operaciones intensivas. Para prevenir el sobrecalentamiento, se pueden añadir disipadores de calor o ventiladores.



REFERENCIA

Abrego, M., y Molinos, S. (s.f.). *Equipos de protección personal*.

Alarcón, J., & Poma, R. (2021). Desarrollo de un algoritmo computacional de detección de equipos de protección eléctrica en personas, orientado a sistemas de vigilancia basados en cámaras IP. https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/657930/Alarcon_CJ.pdf?sequence=3&isAllowed=y

Berzal, F. (s.f.). *Entrenamiento de redes neuronales*. Universidad de Granada. Granada.

Blasco Herreiz, J. (2018). Técnicas de estadística computacional para visión por computador. UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA.

Castillo, C. (2014). *Dirección automático de movimientos corporales mediante técnicas de procesamiento digital de imágenes*. Piura: Universidad Nacional de Piura. Piura: Universidad Nacional de Piura.

Domínguez Pavón, S. (2019). Identificación del modelo de cámara mediante Redes Neuronales Convolucionales. Sevilla, España: Dep. Teoría de la Señal y Comunicaciones Escuela Técnica Superior de Ingeniería Universidad de Sevilla.

ELINUX. (2020). *elinux.org*. Recuperado el 30 de 11 de 2021, de https://elinux.org/RPi_raspi-config

Estela, J. (2016). *Análisis Comparativo De Algoritmos De Reconocimiento De Imágenes Por Descriptores De Color Para La Identificación De Billetes*. Chiclayo: Universidad Señor de Sipan. Chiclayo - Perú.

IEEE Signal Processing Society. (s.f.). *signalprocessingsociety.org*. Obtenido de <https://signalprocessingsociety.org/>

Jiménez, M. (2015). *Desarrollo de un sistema de visión artificial para la detección de aglomeración de personas en un semáforo*. Loja - Ecuador: Universidad Nacional de Loja. Loja - Ecuador.

Kaggle. (2017). Forensic Camera Model Identification Challenge. IEEE's Signal Processing Society - Camera Model Identification. Obtenido de <https://www.kaggle.com/c/sp-society-camera-model-identification>

Kurosawa, K., Kuroki, K., y Saitoh, N. (2000). CCD fingerprint method-identification of a video camera from videotaped images. Proceedings of IEEE International Conference on Image Processing.

Marr, D. (1982). *Visión*. San Francisco: Freeman.

Martinez, J. (2018). *Sistema de Visión Artificial para la Detección y Corrección de Posturas en Ejercicios realizados por Fisicoculturistas*. Atlacomulco: Universidad Autónoma del estado de México. México.

Martin, A. (2023). Desarrollo de un sistema de visión artificial basado en redes convolucionales para el conteo automatizado de microcápsulas en tejidos. <https://riunet.upv.es/bitstream/handle/10251/198510/Martin%20-%20Desarrollo%20de%20un%20sistema%20de%20vision%20artificial%20>

basado%20en%20redes%20convolucionales%20para%20el%20co....pdf?s
equence=1&isAllowed=y

Massiris, M., Fernández, J., y Delrieux, C. (2018). *Detección de equipos de protección personal mediante red neuronal convolucional YOLO*. ResearchGate.

Obregón, Á. (2017). *Implementación de sistema de video cámaras IP como medio de seguridad para el Tecnológico*. Mexico: Universidad Autónoma de Yucatán. Yucatán - Mexico.

P., D., y P., S. (2019). *MicroPython Documentation*. Release 1.10.

Palomino, J. (2020). *Factores de adherencia y uso de equipos de protección personal en el contexto del Covid-19 en enfermeras de un hospital nacional, 2020*. Lima: Universidad César Vallejo.

Pichucho, J., Gordon, A., y Mendoza, D. (2018). *Interpretación de lenguaje de señas ecuatoriano empleando visión por computador*. *Revista Ibérica de Sistemas e Tecnologías de Informação Iberian Journal of Information Systems and Technologies*.

Purizaca, J. (2019). *Diseño de laboratorio para pruebas de aislamiento eléctrico a equipos de protección personal para garantizar los trabajos según norma ISO/ IEC 17025:2006 - Chiclayo*. Chiclayo: Universidad César Vallejo.

Ramanath, R., Snyder, W., y Yoo, Y. (2005). Color Image Processing Pipeline in Digital Still Cameras. *Vol. 22*. IEEE Signal Processing Magazine.

Raspbian. (2021). *raspbian.org*. Recuperado el 30 de 11 de 2021, de <https://www.raspbian.org/RaspbianAbout>

Rojas, J., y Trujillo, R. (2018). *Algoritmo meta-heurístico Firefly aplicado al pre-entrenamiento de redes neuronales artificiales*. La Habana, Cuba: *Universidad de las Ciencias Informáticas*. La Habana - Cuba.

Seená, y Sudershan. (2019). *SMART HEALTHCARE MONITORING SYSTEM USING RASPBERRY Pi ON iot PLATFORM*. *ARNP Journal of Engineering and Applied Sciences*.

Stamm, M., Bestagini, P., y Marcenaro, L. (2018). *Forensic Camera Model Identification*. Highlights from the IEEE Signal Processing Cup 2018 Student Competition.

Sucar, E. (s.f.). *Visión Computacional*. Puebla, México: *Instituto Nacional de Astrofísica, Óptica y Electrónica*.

Vargas, F. (2015). *Implementación de un sistema inteligente para reconocimiento de señales preventivas de seguridad vial*. Trujillo: *Universidad Nacional de Trujillo*. Trujillo - Perú.

Veónica, y Angelo. (2017). *La programación de ordenadores. Reflexiones sobre la necesidad de un abordaje interdisciplinar*. España: *Universidad Autónoma de Madrid*.

Vilca, C. (2019). *Diseño e implementación de sistema de visión artificial con una cámara de escaneo lineal para la inspección de etiquetas en envases cilíndricos*. Arequipa: *Universidad Católica de Santa María*. Arequipa - Perú.

ANEXOS

Anexo 1. Cronograma

Tabla 13
Cronograma

	Abril		Mayo			Junio		
	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6	Semana 7	Semana 8
Planteamiento y Marco Conceptual	X	X						
Estudio y modelamiento del algoritmo		X	X	X				
Programación en Python				X	X			
Pruebas y mejoras para el reconocimiento de EPP					X	X		
Cálculo del tiempo de respuesta del SVA							X	
Desarrollo de conclusiones y recomendaciones								X

Nota. Elaboración propia

Anexo 2. Costo de los materiales principales

CanaKit Raspberry Pi 3 B+ (B Plus)

amazon Perú

Estamos entregando a tu región con opciones de envío limitadas. Por favor anticipa un periodo de entrega prolongado. Más información.

Canakit Raspberry Pi 3 B+ (B Plus) con funda transparente premium y fuente de alimentación de 2,5 A

2,369 calificaciones | 136 preguntas respondidas

Amazon's Choice para "raspberrypi 3"

Precio: US\$ 59.99 + US\$ 42.30 de envío y depósito de derechos de importación a Perú Detalles

Marca Canakit

Tamaño de la memoria de 1 GB la computadora

Tecnología de comunicación inalámbrica Bluetooth

Interfaz de hardware USB, Ethernet

Sobre este artículo

- Incluye Raspberry Pi 3 B+ (B Plus) con procesador Quad-Core de 64 bits de 1,4 GHz y 1 GB de RAM
- CanaKit - Fuente de alimentación USB de 2,5 A con cable micro USB y filtro de ruido, especialmente diseñado para Raspberry Pi 3 B+ (listado UL)
- LAN inalámbrica de doble banda de 2,4 GHz y 5 GHz IEEE 802.11 b/g/n/ac, capacidad Ethernet mejorada
- Funda transparente premium, juego de 2 disipadores de calor de aluminio.
- Guía de inicio rápido de CanaKit

Cámara Web 2560P 2K Full Hd con Micrófono Oem-Negro

Cámara Web 2560P 2K Full Hd

linio.com.pe/cama-mara-web-2560p-2k-full-hd-con-micro-fono-oem-neg-o-j05j62?sku=0E991E11M1YY1PE&sr=8aad14~v2_oaUTXrIAKIAa5SR85PwTJK/PPwSd~Jwl_c_hNVHTiw3zSteRDXRV...

Categorías

Solic hoy Promos para ti Veinde en Linio Ayuda Envío Gratis Imcalibres Ganeas CMR

Inicio / Computador / Accesorios de computadores / Webcams / Cámara Web 2560P 2K Full Hd con Micrófono Oem-Negro

Cámara Web 2560P 2K Full Hd con Micrófono Oem-Negro

Marca OEM 4.42 reseñas

US\$ 280.00 - 65%

S/ 85.90

S/ 82.90

Acumula hasta 82 CMR Puntos

Recíbelo Mañana
Recíbelo el 17 de enero Gratis en Lima, San Isidro

Recogelo gratis en tienda
[Calcular envío en otra dirección](#)

1

CMR Pasa tu Tarjeta CMR Visa aquí y disfruta un mundo de beneficios

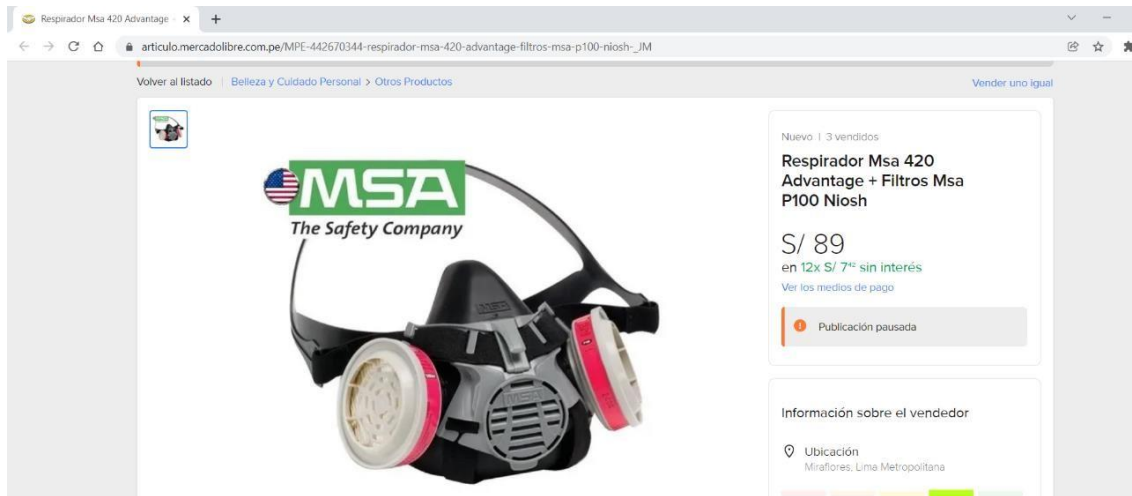
¡Tus compras están seguras! Devoluciones gratuitas

Características destacadas

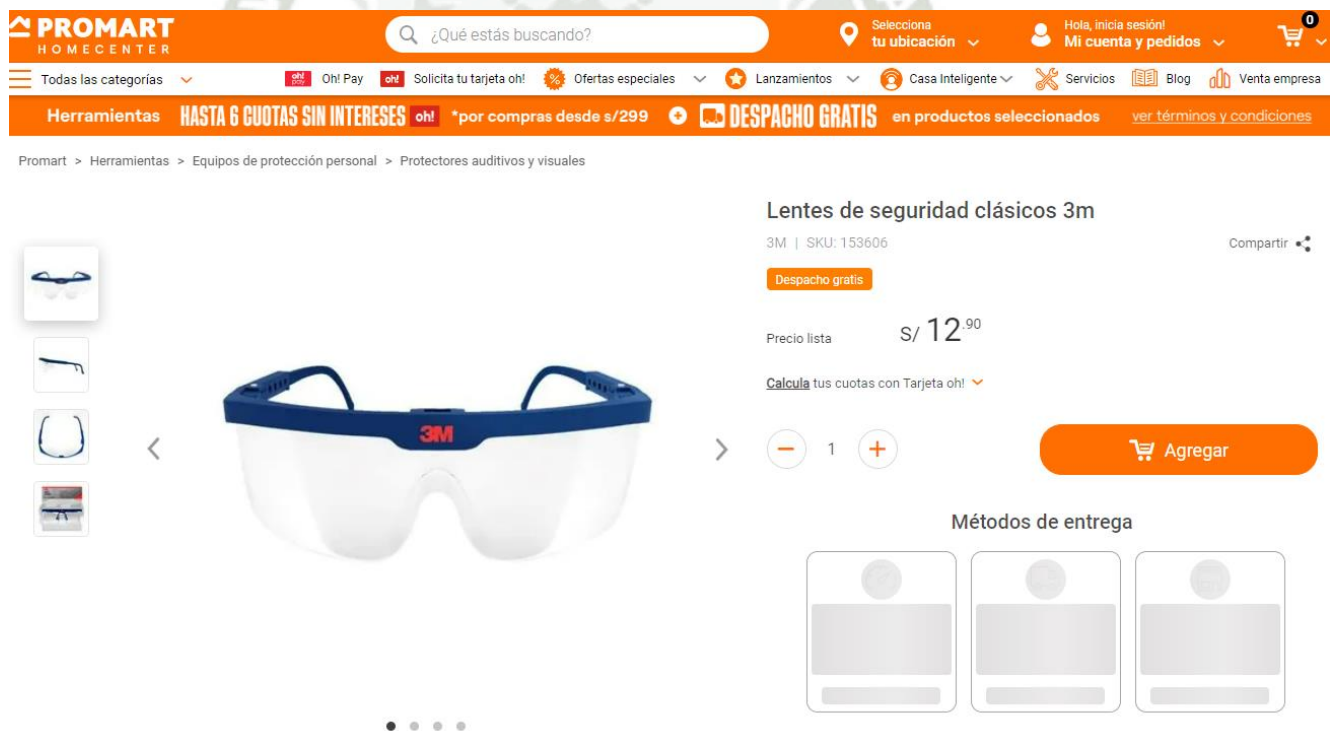
Despachamos todos los días para que el producto te llegue lo más rápido posible!

- Cámara de computadores HD 1080P y 2K 2560P
- Cámara de la computadora, webcam
- Oem Micrófono

Respirador Msa 420 Advantage + Filtros Msa P100 Niosh



Lentes de seguridad clásicos 3m



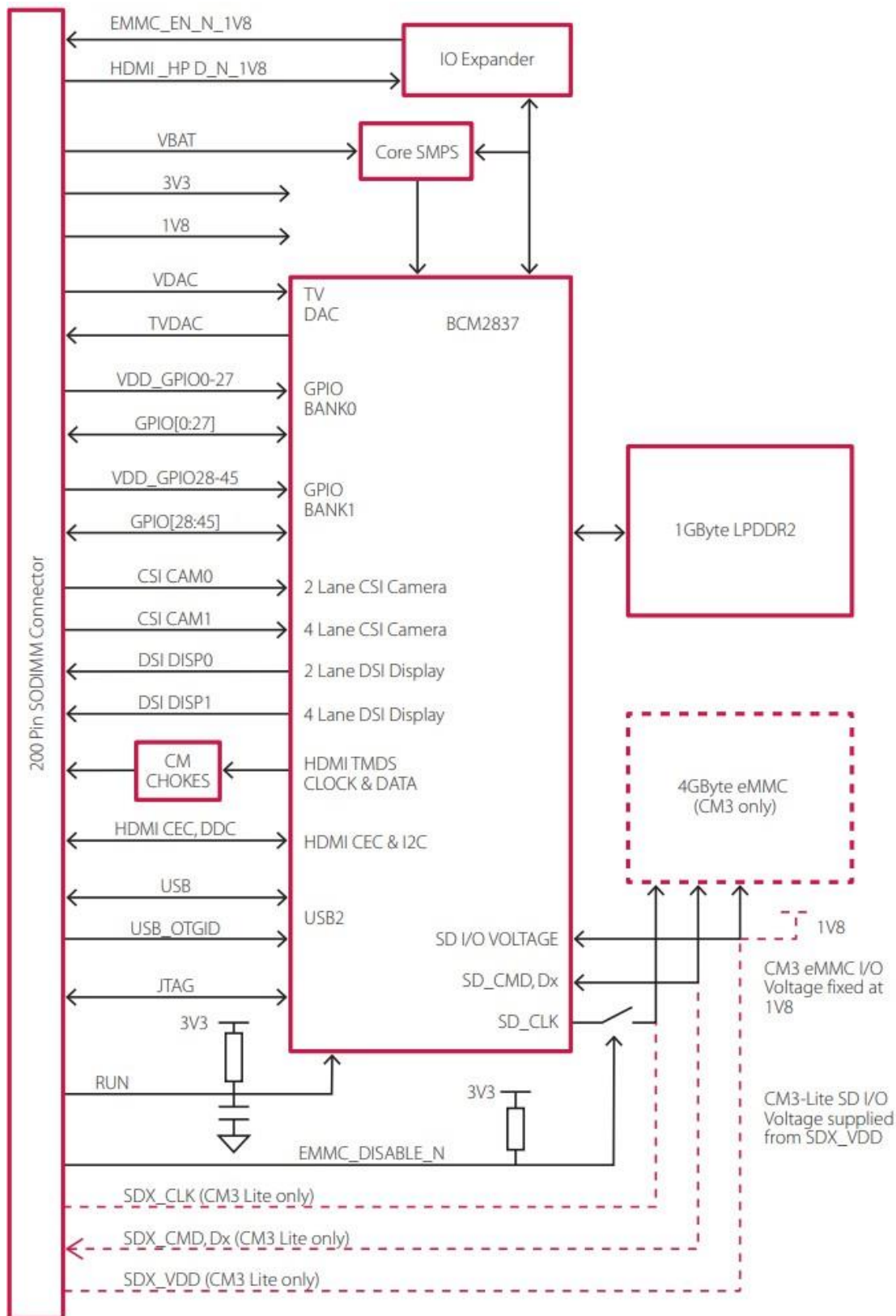
Casco con Ratchet Amarillo

The screenshot shows the SODIMAC website interface. At the top, there is a search bar with the text "¿Qué estás buscando?", a location indicator for "San Miguel", and a shopping cart icon. Below the search bar, there are navigation tabs for various product categories: BAÑO, COCINA Y LIMPIEZA; AIRE LIBRE, JARDÍN Y PARRILLAS; AUTOMÓVIL; CONSTRUCCIÓN Y FERRETERÍA; DECORACIÓN, MENAJE E ILUMINACIÓN; ELECTROHOGAR, TECNOLOGÍA Y CLIMATIZACIÓN; and PROYECTOS E INSPIRACIÓN. The main content area displays a yellow Ratchet helmet. To the right of the helmet, the product details are shown: "Casco con Ratchet Amarillo", "Código 1349457", a star rating of 0, and a price of "S/ 15.90 C/U". There is a note that says "Pague menos desde 10 unidades". Below the price, there is a promotional banner for "CMR VISA" with the text "Obtén tu CMR VISA ¡Y disfruta un mundo de beneficios!" and a "Solicítala aquí" button. A quantity selector shows "1" and an "Agregar al carro" button. At the bottom of the product details, there is a "Satisfacción Garantizada" section with a gear icon and the text: "Si este producto no cumple con tus expectativas tienes 10 días desde su recepción para devolverlo en cualquiera de nuestras tiendas o llamando al (01) 203 0420 opción 4". Below this, there is a section for "Opciones de entrega para San Miguel".



Anexo 3. Datasheet RASPBERRY Pi 3B+

DIAGRAMA DE BLOQUES (CM3/CM3L)



ASIGNACIÓN DE PINES

Pin Name	DIR	Voltage Ref	PDN ^o State	If Unused	Description/Notes
RUN and Boot Control (see text for usage guide)					
RUN	I	3V3 ^b	Pull High	Leave open	Has internal 10k pull up
EMMC_DISABLE_N	I	3V3 ^b	Pull High	Leave open	Has internal 10k pull up
EMMC_EN_N_1V8	O	1V8	Pull High	Leave open	Has internal 2k2 pull up
GPIO					
GPIO[27:0]	I/O	GPIO0-27_VDD	Pull or Hi-Z ^c	Leave open	GPIO Bank 0
GPIO[45:28]	I/O	GPIO28-45_VDD	Pull or Hi-Z ^c	Leave open	GPIO Bank 1
Primary SD Interface^{d,e}					
SDX_CLK	O	SDX_VDD	Pull High	Leave open	Primary SD interface CLK
SDX_CMD	I/O	SDX_VDD	Pull High	Leave open	Primary SD interface CMD
SDX_Dx	I/O	SDX_VDD	Pull High	Leave open	Primary SD interface DATA
USB Interface					
USB_Dx	I/O	-	Z	Leave open	Serial interface
USB_OTGID	I	3V3		Tie to GND	OTG pin detect
HDMI Interface					
HDMI_SCL	I/O	3V3 ^b	Z ^f	Leave open	DDC Clock (5.5V tolerant)
HDMI_SDA	I/O	3V3 ^b	Z ^f	Leave open	DDC Data (5.5V tolerant)
HDMI_CEC	I/O	3V3	Z	Leave open	CEC (has internal 27k pull up)
HDMI_CLKx	O	-	Z	Leave open	HDMI serial clock
HDMI_Dx	O	-	Z	Leave open	HDMI serial data
HDMI_HPD_N_1V8	I	1V8	Pull High	Leave open	HDMI hotplug detect
CAM0 (CSI0) 2-lane Interface					
CAM0_Cx	I	-	Z	Leave open	Serial clock
CAM0_Dx	I	-	Z	Leave open	Serial data
CAM1 (CSI1) 4-lane Interface					
CAM1_Cx	I	-	Z	Leave open	Serial clock
CAM1_Dx	I	-	Z	Leave open	Serial data
DSI0 (Display 0) 2-lane Interface					
DSI0_Cx	O	-	Z	Leave open	Serial clock
DSI0_Dx	O	-	Z	Leave open	Serial data
DSI1 (Display 1) 4-lane Interface					
DSI1_Cx	O	-	Z	Leave open	Serial clock
DSI1_Dx	O	-	Z	Leave open	Serial data
TV Out					
TVDAC	O	-	Z	Leave open	Composite video DAC output
JTAG Interface					
TMS	I	3V3	Z	Leave open	Has internal 50k pull up
TRST_N	I	3V3	Z	Leave open	Has internal 50k pull up
TCK	I	3V3	Z	Leave open	Has internal 50k pull up
TDI	I	3V3	Z	Leave open	Has internal 50k pull up
TDO	O	3V3	O	Leave open	Has internal 50k pull up

ESPECIFICACIONES ELÉCTRICAS

Symbol	Parameter	Minimum	Maximum	Unit
V _{BAT}	Core SMPS Supply	-0.5	6.0	v
3V3	3V3 Supply Voltage	-0.5	4.10	v
1V8	1V8 Supply Voltage	-0.5	2.10	v
VDAC	TV DAC Supply	-0.5	4.10	v
GPIO0-27_VDD	GPIO0-27 I/O Supply Voltage	-0.5	4.10	v
GPIO28-45_VDD	GPIO28-27 I/O Supply Voltage	-0.5	4.10	v
SDX_VDD	Primary SD/eMMC Supply Voltage	-0.5	4.10	v

Table 4: Absolute Maximum Ratings

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V _{IL}	Input low voltage ^a	VDD_IO = 1.8V VDD_IO = 2.7V	- -	- -	0.6 0.8	V V
V _{IH}	Input high voltage ^a	VDD_IO = 1.8V VDD_IO = 2.7V	1.0 1.3	- -	- -	V V
I _{IL}	Input leakage current	TA = +85°C	-	-	5	μA
C _{IN}	Input capacitance	-	-	5	-	pF
V _{OL}	Output low voltage ^b	VDD_IO = 1.8V, IOL = -2mA VDD_IO = 2.7V, IOL = -2mA	- -	- -	0.2 0.15	V V
V _{OH}	Output high voltage ^b	VDD_IO = 1.8V, IOH = 2mA VDD_IO = 2.7V, IOH = 2mA	1.6 2.5	- -	- -	V V
I _{OL}	Output low current ^c	VDD_IO = 1.8V, VO = 0.4V VDD_IO = 2.7V, VO = 0.4V	12 17	- -	- -	mA mA
I _{OH}	Output high current ^c	VDD_IO = 1.8V, VO = 1.4V VDD_IO = 2.7V, VO = 2.3V	10 16	- -	- -	mA mA
R _{PU}	Pullup resistor	-	50	-	65	kΩ
R _{PD}	Pulldown resistor	-	50	-	65	kΩ

^a Hysteresis enabled

^b Default drive strength (8mA)

^c Maximum drive strength (16mA)

Table 5: DC Characteristics

Pin Name	Symbol	Parameter	Minimum	Typical	Maximum	Unit
Digital outputs	t _{rise}	10-90% rise time ^a	-	1.6	-	ns
Digital outputs	t _{fall}	90-10% fall time ^a	-	1.7	-	ns
GPCLK	t _{JOSC}	Oscillator-derived GPCLK cycle-cycle jitter (RMS)	-	-	48	ps
GPCLK	t _{JPLL}	PLL-derived GPCLK cycle-cycle jitter (RMS)	-	-	20	ps

^a Default drive strength, CL = 5pF, VDD IOx = 3.3V


Table 6: Digital I/O Pin AC Characteristics

Anexo 4. Hoja de especificaciones técnicas del prototipo

Este anexo detalla las especificaciones técnicas del prototipo de sistema de detección de equipos de protección personal (EPP) mediante visión artificial implementado para supervisar a trabajadores del sector industrial.

Hardware Utilizado

	Descripción	Versión	Características
 <p>Raspberry Pi 3 B+</p>	Microcomputadora de bajo costo y alto rendimiento utilizada como unidad central de procesamiento del sistema.	Modelo 3 B+	<ul style="list-style-type: none"> - Procesador: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4 GHz - Memoria RAM: 1 GB LPDDR2 SDRAM - Almacenamiento: MicroSD - Conectividad: WiFi 802.11 b/g/n/ac, Bluetooth 4.2, Ethernet - Puertos: 4 x USB 2.0, 1 x HDMI, 1 x Audio estéreo y video compuesto, GPIO
 <p>Camera 2K HD</p>	Cámara de alta definición para capturar imágenes y video, conectada a la Raspberry Pi para la captura de datos visuales.	Logitech 2K HD	<ul style="list-style-type: none"> - Resolución: 2K (2048 x 1080 píxeles) - Interfaz: CSI (Camera Serial Interface) - FPS: Hasta 30 cuadros por segundo - Características adicionales: Autoenfoco, balance de blancos automático
 <p>Monitor con cable HDMI</p>	Monitor externo para visualizar la salida de video del sistema durante las pruebas y el desarrollo.	Modelo 20MK400 H B	<ul style="list-style-type: none"> - Resolución: Full HD (1920 x 1080 píxeles) - Conectividad: HDMI - Tamaño: 19" - Entrada: 100-240 Vca, 50/60 Hz
<p>Teclado y mouse USB</p>	Periféricos de entrada utilizados para interactuar con el sistema durante su	Marca Logitech	<ul style="list-style-type: none"> - Teclado: Conexión USB, diseño QWERTY estándar - Mouse: Conexión USB, óptico con dos botones y rueda de desplazamiento

	configuración y operación.		
---	----------------------------	--	--

Software Utilizado

	Descripción	Versión	Características
Sistema Operativo: Raspbian OS	Sistema operativo basado en Debian, optimizado para el hardware de Raspberry Pi.	Debian 9 (stretch)	- Interfaz gráfica de usuario (GUI) - Soporte para múltiples lenguajes de programación - Herramientas de desarrollo preinstaladas
Lenguaje de Programación: Python	Lenguaje de programación de alto nivel utilizado para el desarrollo de la aplicación de detección de EPP.	Python 3.7	- Sintaxis clara y fácil de aprender - Amplia colección de bibliotecas y frameworks
Librería OpenCV	Biblioteca de código abierto para visión artificial y procesamiento de imágenes.	OpenCV 4.x	- Detección y reconocimiento de objetos - Manipulación y análisis de imágenes y video
Librería MediaPipe	Framework para el procesamiento de medios y aprendizaje automático.	v0.10.11	- Detección de poses y gestos - Modelos preentrenados para visión artificial
Librería Numpy	Biblioteca para el manejo de arreglos y matrices multidimensionales.	v1.11	- Operaciones matemáticas y estadísticas - Soporte para grandes volúmenes de datos

Configuración y Funcionamiento del Sistema

- Instalación del Sistema Operativo

Descargar la imagen de Raspbian OS desde el sitio oficial de Raspberry Pi.

Grabar la imagen en una tarjeta MicroSD utilizando la herramienta Balena Etcher.
Insertar la tarjeta MicroSD en la Raspberry Pi y completar el proceso de configuración inicial

- **Configuración del Entorno de Desarrollo**

Instalar Python y las librerías necesarias:

```
sudo apt-get update  
  
sudo apt-get install python3 python3-pip  
  
pip3 install opencv-python mediapipe numpy
```

Configurar la cámara 2K HD conectándola a la interfaz CSI de la Raspberry Pi.

- **Desarrollo del Software**

Escribir el código en Python para la captura de imágenes y video utilizando OpenCV

Utilizar MediaPipe para la detección de EPP en tiempo real.

Procesar y analizar las imágenes con Numpy para obtener resultados precisos.

- **Pruebas y Validación**

Realizar pruebas en un entorno simulado representando un área de trabajo industrial.

Evaluar la precisión y eficiencia del sistema en la detección de EPP.

Ajustar parámetros y algoritmos según sea necesario para mejorar el rendimiento.

Nota: Todos estos procedimientos se detallan en el Capítulo III.

Anexo 5. Códigos en Python

Código: Generador de base de datos

```
import cv2
import numpy as np
import imutils
import os

Datos = 'nombre_carpeta'
if not os.path.exists(Datos):
    print('Carpeta creada: ',Datos)
    os.makedirs(Datos)

cap = cv2.VideoCapture(0)

x1, y1 = 190, 80
x2, y2 = 450, 340

count = 0
while True:
    ret, frame = cap.read()
    if ret == False: break
    frame = cv2.flip(frame, 1)
    imAux = frame.copy()
    cv2.rectangle(frame, (x1,y1), (x2,y2), (255,0,0), 2)

    objeto = imAux[y1:y2,x1:x2]
    objeto = imutils.resize(objeto,width=72)
    #print(objeto.shape)

    k = cv2.waitKey(1)
    if k == ord('s'):
```

```
cv2.imwrite(Datos+'/objeto_{}.jpg'.format(count),objeto)
print('Imagen guardada:'+'{/objeto_{}.jpg'.format(count))
count = count +1

if k == 27:
    break

cv2.imshow('frame',frame)
cv2.imshow('objeto',objeto)

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

Código: Entrenamiento del modelo

```
Import cv2
import os
import numpy as np

# //////////// Importar los fotogramas generados ////////////
dataPath = "/home/pi/Desktop/Accesorios/Dataset_faces"
dir_list = os.listdir(dataPath)

print("Lista archivos:", dir_list)

labels = []

facesData = []
label = 0

for name_dir in dir_list:
    dir_path = dataPath + "/" + name_dir
    for file_name in os.listdir(dir_path):
        image_path = dir_path + "/" + file_name
        print(image_path)

        image = cv2.imread(image_path, 0)
        #cv2.imshow("Image", image)
```

```
#cv2.waitKey(10)
facesData.append(image)
labels.append(label)

label += 1

print("Etiqueta 0: ", np.count_nonzero(np.array(labels) == 0))
print("Etiqueta 1: ", np.count_nonzero(np.array(labels) == 1))
print("Etiqueta 2: ", np.count_nonzero(np.array(labels) == 2))
# LBPH FaceRecognizer
face_mask = cv2.face.LBPHFaceRecognizer_create()
# Entrenamiento

print("Entrenando...")
face_mask.train(facesData, np.array(labels))
# Almacenar modelo
face_mask.write("face_accessories_model.xml")
print("Modelo almacenado")
```

Código: Sistema de detección de EPP

```
import cv2
import os
import mediapipe as mp

mp_face_detection = mp.solutions.face_detection
LABELS = ["Sin_accesorios", "Con_lentes", "Con_casco"]
# Leer el modelo
face_mask = cv2.face.LBPHFaceRecognizer_create()
face_mask.read("face_accessories_model.xml")

cap = cv2.VideoCapture(0)

with mp_face_detection.FaceDetection(
    min_detection_confidence=0.5) as face_detection:
    while True:
        ret, frame = cap.read()
        if ret == False: break
        frame = cv2.flip(frame, 1)
```

```

height, width, _ = frame.shape
frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
results = face_detection.process(frame_rgb)

if results.detections is not None:
    for detection in results.detections:
        xmin =
int(detection.location_data.relative_bounding_box.xmin * width)
        ymin =
int(detection.location_data.relative_bounding_box.ymin * height)
        w =
int(detection.location_data.relative_bounding_box.width * width)
        h =
int(detection.location_data.relative_bounding_box.height * height)
        if xmin < 0 and ymin < 0: continue
        #cv2.rectangle(frame, (xmin, ymin), (xmin + w, ymin +
h), (0, 255, 0), 5)

        face_image = frame[ymin : ymin + h, xmin : xmin + w]
        face_image = cv2.cvtColor(face_image,
cv2.COLOR_BGR2GRAY)

        face_image = cv2.resize(face_image, (72, 72),
interpolation=cv2.INTER_CUBIC)

        result = face_mask.predict(face_image)
        #cv2.putText(frame, "{}".format(result), (xmin, ymin
- 5), 1, 1.3, (210, 124, 176), 1, cv2.LINE_AA)

        if result[1] < 150:
            color = (0, 0, 255) if LABELS[result[0]]==
"Sin accesorios" else (0, 255, 0)
            cv2.putText(frame,
"{}".format(LABELS[result[0]]), (xmin, ymin - 15), 2, 1, color, 1,
cv2.LINE_AA)

            cv2.rectangle(frame, (xmin, ymin), (xmin + w,
ymin + h), color, 2)

        cv2.imshow("Frame", frame)
        k = cv2.waitKey(1)
        if k == 27:
            break

cap.release()
cv2.destroyAllWindows()

```

Código: Obtención de la matriz de confusión

```
from sklearn.metrics import confusion_matrix
from string import ascii_uppercase
import numpy as np
import pandas as pd
import seaborn as sns

# Etiquetamos a los accesorios de la siguiente manera:
# Casco = 1
# Lentes = 2
# Mascarilla = 3
# Colocamos las etiquetas verdaderas y las predicciones como ejemplo en []
y_verd = np.array([1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,3,3,3,3,3,2,3,3,3,3,3,3,3])
# Etiquetas verdaderas
y_pred = np.array([1,1,1,1,1,1,1,1,1,2,2,3,2,2,2,2,3,3,3,3,2,3,3,3,3,3,3,3])
# Predicciones
# Calculo de la matriz de confusión
confm = confusion_matrix(y_verd, y_pred)
# Normalizacion de la matriz de confusión para mostrar los resultados en
# porcentajes
confm_norm = confm.astype('float') / confm.sum(axis=1)[:, np.newaxis]
# Definimos las etiquetas personalizadas para las clases verdaderas y las
# predicciones
etiquetas = ['Casco', 'Lentes', 'Mascarilla']
# Creamos un DataFrame a partir de la matriz de confusión normalizada con
# las etiquetas personalizadas
df_cm = pd.DataFrame(confm_norm, index=etiquetas, columns=etiquetas)
# Creamos una visualización de la matriz de confusión normalizada como un
# mapa de calor en tonos de azul
grafica = sns.heatmap(df_cm, cmap='Blues', annot=True, fmt='.2f')
# Establecemos etiquetas personalizadas para los ejes x e y
grafica.set(xlabel='Predict label', ylabel='True label')
```