



**UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE INGENIERÍA INDUSTRIAL
CARRERA DE INGENIERÍA EN TELEINFORMÁTICA**

**TRABAJO DE TITULACIÓN
PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN TELEINFORMÁTICA**

**ÁREA
TECNOLOGÍAS APLICADAS**

**TEMA
“DESARROLLO DE UN SISTEMA CON INTELIGENCIA
ARTIFICIAL PARA EL PROCESAMIENTO DE VÍDEO EN
LA DETECCIÓN DEL USO DE EQUIPOS DE SEGURIDAD
PERSONAL DE LOS TÉCNICOS DEL LABORATORIO
CLÍNICO VIDASSAN DE LA CIUDAD DE GUAYAQUIL”**

**AUTORA
VÁSQUEZ HUNGRÍA ROSA MARIAM**

**DIRECTOR DEL TRABAJO
ING. MEC. ARÁUZ ARROYO OSWALDO ORLANDO, MG**

GUAYAQUIL, SEPTIEMBRE 2022



**ANEXO XI.- FICHA DE REGISTRO DE
TRABAJO DE TITULACIÓN
FACULTAD INGENIERÍA INDUSTRIAL
CARRERA INGENIERÍA INDUSTRIAL**



REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA			
FICHA DE REGISTRO DE TRABAJO DE TITULACIÓN			
TÍTULO:	DESARROLLO DE UN SISTEMA CON INTELIGENCIA ARTIFICIAL PARA EL PROCESAMIENTO DE VÍDEO EN LA DETECCIÓN DEL USO DE EQUIPOS DE SEGURIDAD PERSONAL DE LOS TÉCNICOS DEL LABORATORIO CLÍNICO VIDASSAN DE LA CIUDAD DE GUAYAQUIL		
AUTOR (apellidos y nombres):	VÁSQUEZ HUNGRÍA ROSA MARIAM		
TUTOR y REVISOR (apellidos y nombres):	ING. ARÁUZ ARROYO OSWALDO ORLANDO, MG ING. PLAZA VARGAS ANGEL MARCEL, MG		
INSTITUCIÓN:	UNIVERSIDAD DE GUAYAQUIL		
UNIDAD/FACULTAD:	FACULTAD DE INGENIERÍA INDUSTRIAL		
MAESTRÍA/ESPECIALIDAD:			
GRADO OBTENIDO:	INGENIERA EN TELEINFORMÁTICA		
FECHA DE PUBLICACIÓN:	27 DE SEPTIEMBRE DE 2022	No. DE PÁGINAS:	124
ÁREAS TEMÁTICAS:	TECNOLOGÍAS APLICADAS		
PALABRAS CLAVES/ KEYWORDS:	Equipos de protección personal, aprendizaje profundo, detección en tiempo real, LabelImg, SSD MobileNetV2		
<p>RESUMEN: En busca de reducir los riesgos de propagación de enfermedades por una pandemia vigente, los técnicos de laboratorio que están en contacto con personas contagiadas de este virus y de otros más, demuestra cuán primordial es el uso de los equipos de protección personal para no poner en riesgo su integridad física y biológica, la inspección periódica de los colaboradores para verificar sus usos fue la causa por la que se propuso desarrollar un sistema de detección de equipos de protección personal en tiempo real con inteligencia artificial, el cual detecta no solo el uso de dos de los tipos más importantes, así como los guantes y mascarilla, sino a su vez la falta de los mismos, con la aplicación de un modelo pre-entrenado como lo es SSD MobileNetV2, usando un dataset personalizado de imágenes con diferentes ángulos de los objetos para un mejor resultado, empleando la herramienta Tensorflow. Con el sistema entrenado se demuestra su funcionamiento al generar notificaciones en la pantalla cuando el equipamiento está completo o incompleto.</p> <p>ABSTRACT: In order to reduce the risk of spreading diseases due to a current pandemic, laboratory technicians who are in contact with people infected with this virus and others, shows how important is the use of personal protective equipment to avoid putting at risk their physical and biological integrity, the periodic inspection of employees to verify their use was the reason why it was proposed to develop a detection system of personal protective equipment in real time with artificial intelligence, which detects not only the use of two of the most important types, such as gloves and masks, but also the lack of them, with the application of a pre-trained model such as SSD MobileNetV2, using a customized dataset</p>			

of images with different angles of the objects for a better result, using the Tensorflow tool. With the system trained, its operation is demonstrated by generating notifications on the screen when the equipment is complete or incomplete.

ADJUNTO PDF:	SI (X)	NO
CONTACTO CON AUTOR/ES:	Teléfono: 0984755577	E-mail: rosa.vasquezhun@ug.edu.ec
CONTACTO CON LA INSTITUCIÓN:	Nombre: Ing. Ind. Ramón Maquilón Nicola, Mg.	
	Teléfono: 042-658128	
	E-mail: titulacion.ingenieria.industrial@ug.edu.ec	



**ANEXO XII.- DECLARACIÓN DE AUTORÍA
Y DE AUTORIZACIÓN DE LICENCIA GRATUITA
INTRANSFERIBLE Y NO EXCLUSIVA PARA EL
USO NO COMERCIAL DE LA OBRA
CON FINES NO ACADÉMICOS**



**FACULTAD INGENIERÍA INDUSTRIAL
CARRERA INGENIERÍA EN TELEINFORMÁTICA**

**LICENCIA GRATUITA INTRANSFERIBLE Y NO COMERCIAL DE LA OBRA
CON FINES NO ACADÉMICOS**

Yo **VÁSQUEZ HUNGRÍA ROSA MARIAM** con C.C. No. **0952462349**, certifico que los contenidos desarrollados en este trabajo de titulación, cuyo título es **“DESARROLLO DE UN SISTEMA CON INTELIGENCIA ARTIFICIAL PARA EL PROCESAMIENTO DE VÍDEO EN LA DETECCIÓN DEL USO DE EQUIPOS DE SEGURIDAD PERSONAL DE LOS TÉCNICOS DEL LABORATORIO CLÍNICO VIDASSAN DE LA CIUDAD DE GUAYAQUIL”** son de mi absoluta propiedad y responsabilidad, en conformidad al Artículo 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN*, autorizo la utilización de una licencia gratuita intransferible, para el uso no comercial de la presente obra a favor de la Universidad de Guayaquil.

A handwritten signature in blue ink, which appears to read "Mariam Vázquez H.".

VÁSQUEZ HUNGRÍA ROSA MARIAM
C.C.: 0952462349



**ANEXO VII.- CERTIFICADO
PORCENTAJE DE SIMILITUD
FACULTAD DE INGENIERÍA INDUSTRIAL
CARRERA INGENIERÍA EN TELEINFORMÁTICA**



Habiendo sido nombrado **ING. MEC. ARÁUZ ARROYO OSWALDO ORLANDO, MG**, tutor del trabajo de titulación certifico que el presente trabajo de titulación ha sido elaborado por **VÁSQUEZ HUNGRÍA ROSA MARIAM**, con mi respectiva supervisión como requerimiento parcial para la obtención del título de **INGENIERO EN TELEINFORMÁTICA**.

Se informa que el trabajo de titulación: **“DESARROLLO DE UN SISTEMA CON INTELIGENCIA ARTIFICIAL PARA EL PROCESAMIENTO DE VÍDEO EN LA DETECCIÓN DEL USO DE EQUIPOS DE SEGURIDAD PERSONAL DE LOS TÉCNICOS DEL LABORATORIO CLÍNICO VIDASSAN DE LA CIUDAD DE GUAYAQUIL.”**, ha sido orientado durante todo el periodo de ejecución en el programa antiplagio TURNITIN quedando el 3% de coincidencia.

<https://ev.turnitin.com/app/carta/es/?o=1899161705&u=1133714787&s=1&lang=es>



Firmado electrónicamente por:
**OSWALDO
ORLANDO
ARAUZ
ARROYO**

**ING. MEC. ARÁUZ ARROYO OSWALDO ORLANDO, MG.
TUTOR DE TRABAJO DE TITULACIÓN
C.C.: 1001964749
FECHA: 11 DE SEPTIEMBRE DE 2022**



**ANEXO VI. - CERTIFICADO DEL
DOCENTE-TUTOR DEL RABAJO DE TITULACIÓN
FACULTAD DE INGENIERÍA INDUSTRIAL
CARRERA INGENIERÍA EN TELEINFORMÁTICA**



Guayaquil, 13 de septiembre de 2022

Sr (a).

Ing. Annabelle Lizarzaburu Mora, MG.

Director (a) de Carrera Ingeniería en Teleinformática / Telemática

**FACULTAD DE INGENIERÍA INDUSTRIAL DE LA UNIVERSIDAD DE
GUAYAQUIL**

Ciudad. -

De mis consideraciones:

Envío a Ud. el Informe correspondiente a la tutoría realizada al Trabajo de Titulación **DESARROLLO DE UN SISTEMA CON INTELIGENCIA ARTIFICIAL PARA EL PROCESAMIENTO DE VÍDEO EN LA DETECCIÓN DEL USO DE EQUIPOS DE SEGURIDAD PERSONAL DE LOS TÉCNICOS DEL LABORATORIO CLÍNICO VIDASSAN DE LA CIUDAD DE GUAYAQUIL.**, del estudiante **VÁSQUEZ HUNGRÍA ROSA MARIAM** indicando que ha cumplido con todos los parámetros establecidos en la normativa vigente:

- El trabajo es el resultado de una investigación.
- El estudiante demuestra conocimiento profesional integral.
- El trabajo presenta una propuesta en el área de conocimiento.
- El nivel de argumentación es coherente con el campo de conocimiento.

Adicionalmente, se adjunta el certificado de porcentaje de similitud y la valoración del trabajo de titulación con la respectiva calificación.

Dando por concluida esta tutoría de trabajo de titulación, **CERTIFICO**, para los fines pertinentes, que el estudiante está apto para continuar con el proceso de revisión final.

Atentamente,



Firmado electrónicamente por:
**OSWALDO
ORLANDO
ARAUZ
ARROYO**

**ING. MEC. ARÁUZ ARROYO OSWALDO ORLANDO, MG.
TUTOR DE TRABAJO DE TITULACIÓN
C.C.: 1001964749
FECHA: 13 DE SEPTIEMBRE DE 2022**



**ANEXO VIII.- INFORME DEL DOCENTE REVISOR
FACULTAD INGENIERÍA INDUSTRIAL
CARRERA INGENIERÍA EN TELEINFORMÁTICA**



Guayaquil, 23 de septiembre de 2022

Sr (a).

Ing. Annabelle Lizarzaburu Mora, MG.

Director (a) de Carrera Ingeniería en Teleinformática / Telemática

**FACULTAD DE INGENIERÍA INDUSTRIAL DE LA UNIVERSIDAD DE
GUAYAQUIL**

De mis consideraciones:

Envío a Ud. el informe correspondiente a la REVISIÓN FINAL del Trabajo de Titulación **“DESARROLLO DE UN SISTEMA CON INTELIGENCIA ARTIFICIAL PARA EL PROCESAMIENTO DE VÍDEO EN LA DETECCIÓN DEL USO DE EQUIPOS DE SEGURIDAD PERSONAL DE LOS TÉCNICOS DEL LABORATORIO CLÍNICO VIDASSAN DE LA CIUDAD DE GUAYAQUIL”**, de la estudiante **VÁSQUEZ HUNGRÍA ROSA MARIAM**. Las gestiones realizadas me permiten indicar que el trabajo fue revisado considerando todos los parámetros establecidos en las normativas vigentes, en el cumplimiento de los siguientes aspectos:

Cumplimiento de requisitos de forma:

El título tiene un máximo de 34 palabras.

La memoria escrita se ajusta a la estructura establecida.

El documento se ajusta a las normas de escritura científica seleccionadas por la Facultad.

La investigación es pertinente con la línea y sublíneas de investigación de la carrera.

Los soportes teóricos son de máximo 5 años.

La propuesta presentada es pertinente.

Cumplimiento con el Reglamento de Régimen Académico:

El trabajo es el resultado de una investigación.

El estudiante demuestra conocimiento profesional integral.

El trabajo presenta una propuesta en el área de conocimiento.

El nivel de argumentación es coherente con el campo de conocimiento.

Adicionalmente, se indica que fue revisado, el certificado de porcentaje de similitud, la valoración del tutor, así como de las páginas preliminares solicitadas, lo cual indica el que el trabajo de investigación cumple con los requisitos exigidos.

Una vez concluida esta revisión, considero que el estudiante está apto para continuar el proceso de titulación. Particular que comunicamos a usted para los fines pertinentes.

Atentamente,



Firmado electrónicamente por:
**ANGEL MARCEL
PLAZA VARGAS**

ING. COMP. PLAZA VARGAS ANGEL MARCEL, MG..

DOCENTE TUTOR REVISOR

C.C.: 1001964749

FECHA: 23 DE SEPTIEMBRE DE 2022

Dedicatoria

Dedico mi trabajo de tesis a mi familia y amigos. Un sentimiento especial de gratitud a mis amados padres, Ángel y María Hungría, especialmente mi madre cuyas palabras de aliento y empuje por la tenacidad resuenan en mis oídos, así como por todo el esfuerzo que realizó para que lograra culminar esta profesión. A mis hermanas Cecibel y Alexa que nunca se han separado de mi lado y son muy especiales e importantes.

También dedico esta tesis a mis tíos, Roxana, Graciela y Gustavo, a mi abuela Rosa, por su gran apoyo. Siempre apreciaré todo lo que han hecho. Dedico este trabajo y agradezco especialmente a mi mejor amigo y novio Ernesto Arteaga por su ayuda y ánimo que siempre estuvieron no solo en el desarrollo de este proyecto sino a lo largo de estudio de esta carrera.

Agradecimiento

Agradezco a Dios Todopoderoso mi creador, mi pilar fuerte, mi fuente de inspiración, sabiduría, conocimiento y entendimiento. Él ha sido la fuente de mi fuerza a lo largo de este proyecto y en Él he encontrado ánimo.

Agradezco a mi madre por las incansables horas de trabajo para que hoy esté cumpliendo una de mis metas.

A mi tutor de tesis por su ayuda y guía para la elaboración de esta investigación, a la vez a cada uno de mis profesores a lo largo de estos cinco años, gracias por su profesionalismo y vocación.

Índice General

N°	Descripción	Pág.
1	Introducción	1

Capítulo I

El problema

N°	Descripción	Pág.
1.1	Antecedentes de la investigación	2
1.2	Planteamiento del problema	2
1.2.1	Formulación del problema de investigación	3
1.2.2	Sistematización del problema de investigación	3
1.3	Justificación e importancia	4
1.4	Objetivos de la investigación	4
1.4.1	Objetivo general	4
1.4.2	Objetivos específicos	4
1.5	Alcance del proyecto	5
1.5.1	Delimitación física	5
1.5.2	Alcance	5
1.6	Metodología	5

Capítulo II

Marco teórico

N°	Descripción	Pág.
2.1	Riesgos en un laboratorio	7
2.1.1	Riesgo	7
2.1.2	Tipos de peligros en un laboratorio	7
2.1.2.1	Peligros químicos	8
2.1.2.2	Peligros biológicos	8
2.1.2.3	Peligros físicos	9
2.2	Vías de transmisión de infecciones dentro del laboratorio	10
2.2.1.1	A través del aparato respiratorio	11

2.2.1.2	A través de la piel o mucosas	11
2.2.1.3	A través del tubo digestivo	11
2.2.2	Infecciones adquiridas en el laboratorio	11
2.2.2.1	Gripe	12
2.2.2.2	Varicela	12
2.2.2.3	Tuberculosis	12
2.2.2.4	Salmonelosis	12
2.2.2.5	Hepatitis A	12
2.2.2.6	Hepatitis B	12
2.2.2.7	Hepatitis C	12
2.2.2.8	VIH	13
2.2.2.9	Tétano	13
2.2.2.10	COVID-19	13
2.2.3	Seguridad laboral	13
2.2.3.1	Equipos de protección personal	13
2.2.3.2	Bata de laboratorio	14
2.2.3.3	Protección de mano	14
2.2.3.4	Protección para las vistas y el rostro	15
2.2.4	Visión artificial	16
2.2.4.1	Visión artificial en IA	18
2.2.5	Aprendizaje automático	19
2.2.5.1	Aprendizaje supervisado	19
2.2.5.2	Aprendizaje no supervisado	20
2.2.5.3	Aprendizaje por refuerzo	20
2.2.6	Redes neuronales artificiales	20
2.2.7	Redes neuronales convolucionales	21
2.2.7.1	Arquitectura de las redes convolucionales	23

2.2.8	Aprendizaje profundo	26
2.2.9	Medidas de rendimiento	26
2.2.10	Python 3.9.0	27
2.2.10.1	OpenCV	27
2.2.10.2	Pytorch	28
2.2.10.3	Numpy	28
2.2.11	Anaconda	28
2.2.12	Jupyter	29
2.2.13	Google Colab	29
2.2.14	YOLO	30
2.2.14.1	YOLOv5	30
2.2.14.2	MobileNet	31
2.2.14.3	Tensorflow	32
2.2.14.4	Detector de objetos	33
2.2.14.5	LabelImg	33
2.3	Marco conceptual	34
2.3.1	Algoritmo	34
2.3.2	Dataset	34
2.3.3	Anotación de datos	34
2.3.4	Etiquetado de datos	34

Capítulo III

Desarrollo de la propuesta

N°	Descripción	Pág.
3.1	Metodología de investigación	35
3.2	Metodología del proyecto	35
3.2.1	Metodología de desarrollo del proyecto	35
3.3	Enfoque de la investigación	36
3.3.1	Enfoque cualitativo	36

3.3.2	Operacionalización de variables	36
3.4	Población y muestra	36
3.4.1	Población	36
3.4.2	Muestra	36
3.4.3	Procesamiento y análisis	37
3.4.4	Técnicas de recolección de datos	37
3.4.5	Técnicas estadísticas para el procesamiento de la información	37
3.4.6	Análisis de resultados	37
3.5	Propuesta tecnológica	38
3.5.1	Análisis de factibilidad	39
3.5.2	Factibilidad técnica	39
3.5.3	Factibilidad legal	40
3.5.4	Factibilidad económica	40
3.6	Desarrollo de la propuesta	41
3.6.1	Requerimientos funcionales	41
3.6.2	Requerimientos no funcionales	43
3.6.2.1	Comprensible	43
3.6.2.2	Operatividad	44
3.6.3	Requisitos para operatividad del software	44
3.6.4	Sujetos del sistema	45
3.6.5	Diagrama de casos de usos	46
3.6.5.1	Diagrama de caso de uso del reconocimiento de los guantes y mascarilla	46
3.6.5.2	Diagrama de caso de uso del seguimiento de la mascarilla	47
3.6.5.3	Diagrama de caso de uso del seguimiento de los guantes	47
3.6.5.4	Diagrama de caso de uso cuando no se detecta los EPP	48
3.6.6	Desarrollo del sistema	48
3.6.6.1	Proceso de selección de herramienta	48

3.6.6.1.1 Problemas presentados al usar Jupyter Notebook	48
3.6.6.2 Creación del dataset de la recolección de imágenes	50
3.6.6.3 Estandarización de las imágenes	51
3.6.6.4 División del dataset	52
3.6.6.5 Labels	52
3.6.6.6 Selección de la arquitectura para el sistema	53
3.6.6.7 Proceso de entrenamiento de RNC	54
3.6.7 Etapa de pruebas	56
3.6.7.1 Condición cuando se identifican todos los EPP	59
3.6.7.2 Condición cuando no se detecten todos los EPP	61
3.6.7.1 Resultados de matriz de confusión	62
3.7 Conclusiones	64
3.8 Recomendaciones	65
Anexos	67
Bibliografía	98

Índice de Figuras

N° Descripción	Pág.
1. Ubicación geográfica del laboratorio clínico “VIDASSAN”	5
2. Metodología del desarrollo de software incremental	6
3. Evolución de la cultura de seguridad	7
4. Símbolos de productos químicos	8
5. Riesgo biológico de usar sangre en un laboratorio	9
6. Vías de transmisión de infecciones	10
7. Guantes de nitrilo	14
8. Características de las mascarillas quirúrgicas	15
10. Estructura de una red neuronal	21
11. Estructura de red completamente conectada y convolucional	22
12. Arquitectura de las redes convolucionales	23
13. Arquitectura de las redes convolucionales	24
14. Proceso de max pooling	24
15. Página de inicio Python	27
16. Interfaz de inicio del navegador de Anaconda	28
17. Interfaz de trabajo de Jupyter	29
18. Interfaz de bienvenida en Google Colab	29
19. Estructura del funcionamiento de las capas en YOLO	30
20. Comparación del funcionamiento de la familia YOLO	31
21. Capas de arquitectura de MobileNet	32
22. Proceso para detección de objetos	33
23. Interfaz de trabajo en LabelImg	33
25. Segundo caso de uso para el seguimiento de mascarilla	47
26. Tercer caso de uso para el seguimiento de guantes	47
27. Diagrama de caso de uso al no detectar los EPP	48
28. Resultados de train con Jupyter Notebook y YOLO	49

29.	Evidencia de las imágenes del dataset	50
30.	Carpetas de datos en Google Colab	51
31.	Gráfico de pastel de la división del dataset	52
32.	Etiquetas de una foto del dataset	53
33.	Comparación de los modelos pre-entrenados	54
34.	Imagen que genera ruido para el sistema	54
35.	Captura de pantalla de la estructura de un archivo xml	55
36.	Captura de pantalla de una de las pruebas realizadas al sistema.	57
37.	Sistema sin detectar guantes	57
38.	Sistema que tiene poco asertividad para detectar guantes	58
39.	Captura de pantalla del proceso de entrenamiento	58
40.	Resultados de modelo con mejores resultados	59
41.	Diagrama de bloques del sistema propuesto,	60
42.	Interfaz de equipamiento completo	61
43.	Interfaz de equipamiento incompleto	61
44.	Detectar “sin_mascarilla”	62

Índice de Tablas

Nº	Descripción	Pág.
1.	Tipos de guantes según su composición	15
2.	Descripción de los tipos de mascarillas	16
3.	Hardware usado para el desarrollo	39
4.	Software usado para el desarrollo	40
5.	Detalle de los costos para la elaboración del sistema	41
6.	Descripción de un requerimiento funcional del sistema	42
7.	Segundo requerimiento funcional	42
8.	Tercer requerimiento funcional del sistema	43
9.	Exigencias no operacionales del sistema	44
10.	Requisitos de operatividad del software para el sistema	45
11.	Sujetos del sistema de detección de EPP en tiempo real	46
12.	Distribución de imágenes del dataset	52
13.	Matriz de confusión del modelo de mascarillas	62
14.	Matriz de confusión del modelo de sin mascarillas	63
15.	Matriz de confusión del modelo de guantes	63
16.	Matriz de confusión del modelo de sin guantes	63

Índice de Anexos

Nº	Descripción	Pág.
1.	Evidencia de código en Jupyter Notebook	67
2.	Código para ingresar imagen de prueba en Jupyter Notebook	67
3.	Resultados de épocas de entrenamiento	67
4.	Resultados gráficos del entrenamiento en Jupyter Notebook	68
5.	Dataset del entrenamiento en Jupyter Notebook	68
6.	Curva de precisión del sistema en Jupyter Notebook	68
7.	Instalar PyQt5	69
8.	Acceder a Drive	69
8.	Importar librerías	69
10.	Etiquetas de clases	69
11.	Ruta de la carpeta	69
12.	Instalar las funcionalidades dependientes del S.O.	70
13.	Rutas de configuración	70
14.	Leer y escribir archivos tar	70
15.	Modelos dependientes en los directorios	70
16.	Manipulación de rutas locales	70
17.	Ruta de los archivos con rutas paths	71
18.	Manipular archivos y directorios	71
19.	Descargar la herramienta wgte	71
20.	Clonar el modelo Tensorflow	72
21.	Se accede a las carpetas	72
22.	Condicionar el poxis	72
23.	Lista de requerimientos en la clonación de Tensorflow	72
24.	Acceder el sistema con el posix	73
25.	Instalar tensorflow-object-detection-api	73
26.	Verificar las rutas de los archivos	73

27.	Importar Tensorflow y se verifica la versión	74
28.	Obtener la ruta del modelo pre-entrenado	74
29.	Crear el mapa de las clases de mascarilla y guante	74
30.	Abrir el archivo de mapa de etiquetas	74
31.	Pasar variables a scripts de tensorflow	75
32.	Copiar la configuración del modelo en la carpeta de train	75
33.	Importar las librerías Tensorflow, config_util, pipeline_pb2	75
34.	Guardar la ruta del archivo en la variable config	75
35.	Configuración de las imágenes para el entrenamiento	75
36.	Crear una configuración de Pipeline	76
37.	Se abren y predicen las 90 clases del modelo pre-entrenado	76
38.	Convertir el mensaje protobuf en cadena	76
39.	Instalar CUDA	76
40.	Configurar para guardar el modelo en la guía de entrenamiento	76
41.	Acceder a la ruta del modelo	77
42.	Trabajar con los mapas de etiquetas	77
43.	Configurar el modelo pre-entrenado	77
44.	Importar CV2, Numpy, Matplotlib	77
45.	Categoría para mapear las etiquetas	78
46.	Descargar CUDA con la potencia general GPU, de los núcleos de NVIDIA	78
47.	Configuración de la imagen para la detección de los objetos	78
48.	Condiciones según se detecten todos o no de los EPP	79
49.	Resultados del proceso de entrenamiento del sistema en dashboard	80
50.	Proceso de entrenamiento del sistema	80
51.	Resultados de imágenes de prueba	80
52.	Imágenes de la carpeta de la carpeta de train para entrenamiento	81
56.	Formato de las encuestas realizadas	88

57.	Diagrama de pastel del conocimiento sobre el peligro al que se exponen	89
58.	Diagrama de barras de los EPP más usados por los técnicos	89
59.	Diagrama de pastel de la disponibilidad de los EPP	90
60.	Diagrama de pastel del porcentaje de importancia de la bata	90
62.	Diagrama de pastel de la aceptación del sistema de detección de EPP	91
63.	Diagrama de pastel de cómo se desea visualizar la detección de EPP	91
64.	Diagrama de pastel del tipo de aviso que se desea en el sistema	92
65.	Diagrama de pastel del tipo de género de los colaboradores	92
66.	Diagrama de pastel del promedio de la edad de los técnicos	92



ANEXO XIII.- RESUMEN DEL TRABAJO DE TITULACIÓN (ESPAÑOL)

FACULTAD INGENIERÍA INDUSTRIAL
CARRERA INGENIERÍA EN TELEINFORMÁTICA



“DESARROLLO DE UN SISTEMA CON INTELIGENCIA ARTIFICIAL PARA EL PROCESAMIENTO DE VÍDEO EN LA DETECCIÓN DEL USO DE EQUIPOS DE SEGURIDAD PERSONAL DE LOS TÉCNICOS DEL LABORATORIO CLÍNICO VIDASSAN DE LA CIUDAD DE GUAYAQUIL”

Autor: Vásquez Hungría Rosa Mariam

Tutor: Ing. Mec. Aráuz Arroyo Oswaldo Orlando, Mg.

Resumen

En busca de reducir los riesgos de propagación de enfermedades por una pandemia vigente, los técnicos de laboratorio que están en contacto con personas contagiadas de este virus y de otros más, demuestra cuán primordial es el uso de los equipos de protección personal para no poner en riesgo su integridad tanto física como biológica, la inspección periódica de los colaboradores para verificar sus usos fue la causa por la que se propuso desarrollar un sistema de detección de equipos de protección personal en tiempo real con inteligencia artificial, el cual detecta no solo el uso de dos de los tipos más importantes, así como los guantes y mascarilla, sino a su vez la falta de los mismos, con la aplicación de un modelo pre-entrenado como lo es SSD MobileNetV2, usando un dataset personalizado de imágenes con diferentes ángulos de los objetos para un mejor resultado, empleando la herramienta Tensorflow. Con el sistema entrenado se demuestra su funcionamiento al generar notificaciones en la pantalla cuando el equipamiento está completo o incompleto.

Palabras Claves: Equipos de protección personal, aprendizaje profundo, detección en tiempo real, LabelImg, SSD MobileNetV2



ANEXO XIV.- RESUMEN DEL TRABAJO DE TITULACIÓN (INGLÉS)



FACULTAD INGENIERÍA INDUSTRIAL CARRERA INGENIERÍA EN TELEINFORMÁTICA

“DEVELOPMENT OF A SYSTEM WITH ARTIFICIAL INTELLIGENCE FOR VIDEO PROCESSING IN THE DETECTION OF THE USE OF PERSONAL PROTECTION EQUIPMENT OF THE TECHNICIANS OF THE VIDASSAN CLINICAL LABORATORY IN GUAYAQUIL CITY”

Author: Vásquez Hungría Rosa Mariam

Advisor: Ing. Mec. Aráuz Arroyo Oswaldo Orlando, Mg.

Abstract

In order to reduce the risk of spreading diseases due to a current pandemic, laboratory technicians who are in contact with people infected with this virus and others, shows how important is the use of personal protective equipment to avoid putting at risk their physical and biological integrity, the periodic inspection of employees to verify their use was the reason why it was proposed to develop a detection system of personal protective equipment in real time with artificial intelligence, which detects not only the use of two of the most important types, such as gloves and masks, but also the lack of them, with the application of a pre-trained model such as SSD MobileNetV2, using a customized dataset of images with different angles of the objects for a better result, using the Tensorflow tool. With the system trained, its operation is demonstrated by generating notifications on the screen when the equipment is complete or incomplete.

Keywords: Personal protective equipment, deep learning, real time detection, LabelImg, MobileNetV2 SSD

Introducción

La percepción visual es uno de los sentidos más importantes para los seres humanos. Se depende en gran medida de la visión siempre que se interactúa con el entorno: cuando se coge un objeto, cuando se desplaza dentro del entorno evitando chocar con lo que hay en el camino o cuando se reconoce a amigos o conocidos por sus caras. Para todas esas acciones, el reconocimiento y la localización de objetos son esenciales.

En primer lugar, para coger un vaso se debe determinar qué parte de la impresión visual corresponde con el vaso antes de saber dónde se tienen que mover las manos para agarrarlo. Sin embargo, no se piensa en estos pasos básicos de procesamiento visual de forma activa. A pesar de parecer tan fácil para el cerebro humano, sigue siendo un gran reto para los sistemas artificiales, como los robots, que tienen que procesar el contenido de las imágenes a detalle para identificar objetos dentro de esta.

El proceso de identificar objetos en vídeos e imágenes se conoce como reconocimiento de objetos. Este reconocimiento puede utilizarse para acelerar la búsqueda de objetos en la visión por ordenador. La gran parte de los algoritmos existentes sólo abordan un pequeño subconjunto de las diferentes tareas necesarias para comprender una imagen y son muy exigentes en términos de recursos computacionales y tiempo de ejecución. Para reproducir al menos una parte de la capacidad de percepción visual humana, habría que combinar varios algoritmos diferentes.

Hacer que un sistema combinado de este tipo funcione en tiempo real con el hardware actual es un gran reto. Dado que las máquinas no pueden detectar los objetos en una imagen de forma instantánea como los humanos, es realmente necesario que los algoritmos sean rápidos, precisos y que detecten los objetos en tiempo real. En este trabajo se explora un pequeño paso hacia este objetivo entrenando un modelo de red neuronal para aprender a detectar a través de vídeo el uso de equipos de protección personal por los técnicos de un laboratorio clínico.

Para el desarrollo del sistema de inteligencia artificial, en el presente trabajo se usó un lenguaje de programación práctico y versátil, junto con algoritmos de gran utilidad que permitieron que este fuera más eficiente.

Capítulo I

El problema

1.1 Antecedentes de la investigación

En el artículo publicado en la revista cubana de ciencias informáticas se detalla la revisión de los algoritmos en la detección y seguimiento de objetos usando redes profundas para una videovigilancia inteligente, sus investigadores exponen que los algoritmos tipo one-stage presentan una mayor rapidez para detectar los objetos, pero que a la vez presentan problemas para detectar objetos pequeños. Además, por su portabilidad recomiendan usar el algoritmo YOLO (You Only Look Once)

Vajhala, Maddineni y Yeruva (2016) en su investigación sobre la detección de armas en cámaras de vigilancia demuestran la eficacia de una red neuronal entrenada con un dataset de 12830 imágenes, a su vez determinaron cuán fundamental es que estas tengan buena calidad resultando un 83.05% de eficacia.

Durante la investigación realizada por Nakib, Khan, Hasan y Uddin, en su sistema lograron detectar sangre, cuchillo y armas para determinar si ocurrió un crimen o no, con el entrenamiento se redujeron las falsas alarmas haciendo el modelo muy eficiente comparado con otros. Determinaron que, para su modelo, TensorFlow fue la mejor plataforma para la detección de esos objetos. Al haber aumentado 50 imágenes conforme los pasos de entrenamiento lograron obtener un porcentaje de asertividad del 90.2 %.

En la ciudad de Quito los investigadores Benavides y Pazmiño, realizaron un modelo que es capaz de detectar mascarillas faciales de cualquier tipo usando un sistema embebido incorporado en un vehículo durante la pandemia, como resultado obtuvieron un 93,53% de exactitud para detectar a través de imágenes que se captura por el carro, y verificar si tiene o no tiene mascarilla, y guardarlas en una base de datos

En 2019, Jovann Pérez analizó el tráfico vehicular con la ayuda de la visión artificial, al usar la herramienta OpenCV que fue de utilidad para dar seguimiento a varios vehículos e incluso la resolución con la que ingresa el vídeo al sistema, lo que permitió que el procesamiento no sea tan demorado.

1.2 Planteamiento del problema

Dentro de los laboratorios la mayoría de los peligros encontrados se dividen en tres categorías principales: químicos, biológicos o físicos (Moreno & Hidalgo, 2015).

Existe un gran grupo de productos químicos los cuales van desde los productos que se usan para la limpieza, como los detergentes, esterilizantes, aromatizantes, fármacos aerosoles, pinturas y muchos otros más.

Alguno de los productos antes mencionados puede provocar efectos erosivos en la piel cuando se tiene un contacto directo con estos. Exponerse sin ningún tipo de protección o cuidado se considera como una amenaza cuando se emplea estos químicos, o están mal conservados.

El uso o manipulación de este tipo de materiales no es la única forma de lastimar o herir a un técnico de laboratorio, mucho de los procesos de análisis de sustancias necesitan que se pongan al calor, al tener temperaturas tan altas puede ser riesgoso y generar heridas térmicas, de bajo grado, pero que daña la piel, una vez más se denota la importancia del porqué es vital el uso de EPP.

La manipulación de bacterias, virus, sangre, tejidos y/o fluidos corporales en el laboratorio puede generar peligros biológicos potenciales. Todos estos materiales pueden transportar enfermedades o alérgenos peligrosos que podrían poner en riesgo al personal del laboratorio.

Los efectos de las enfermedades y los alérgenos pueden ser inmediatos o manifestarse luego de un tiempo, lo que demuestra que al no usar alguno de los equipos de protección personal de uno de los técnicos de laboratorio podría traer como consecuencia un gran deterioro a su salud, posteriormente contagiar a sus familiares o círculo más cercano con nuevos virus producidos en los laboratorios, y en efecto generar un contagio colectivo. Los técnicos del laboratorio VIDASSAN, no están exentos a estos peligros y a diario están expuestos a estas amenazas.

1.2.1 Formulación del problema de investigación

¿Cómo la visión artificial puede aportar al control del uso correcto de los equipos de seguridad personal de los técnicos del laboratorio “VIDASSAN”?

1.2.2 Sistematización del problema de investigación

¿La visión artificial puede facilitar la detección de determinados objetos en tiempo real y a su vez cuando estos no estén?

1.3 Justificación e importancia

El uso de la visión por computadora puede ser de gran valor para prácticamente todas las industrias al hacer sus sistemas más eficientes y organizados, además de poder supervisar objetos.

Los miembros del equipo técnico de un laboratorio clínico deben ejercer el trabajo usando el equipo de protección personal esto se especifica en el manual de bioseguridad creado por el ministerio de salud pública del Ecuador. (2016)

La relevancia de este estudio radicó en conocer los potenciales riesgos que se tiene al no usar los implementos necesarios dentro del laboratorio clínico. El equipo de protección personal se utiliza para crear una barrera entre la persona y los gérmenes, lo que ayuda a prevenir la propagación de gérmenes en el laboratorio y posterior a esto en su entorno personal, para así evitar que trabajadores del laboratorio contraigan infecciones y además garantizar resultados correctos en los análisis.

El reconocimiento en tiempo real del adecuado equipo es una forma de aplicar medidas de bioseguridad para mantener la seguridad del laboratorio, así como para proteger a los empleados mitigando riesgos con el uso de inteligencia artificial.

1.4 Objetivos de la investigación

1.4.1 Objetivo general

Demostrar la eficiencia de un sistema con inteligencia artificial para el procesamiento de vídeo en la detección del uso de equipos de seguridad de los técnicos de un laboratorio clínico.

1.4.2 Objetivos específicos

- Establecer el marco teórico a partir de un análisis de información sobre el procesamiento de vídeo con inteligencia artificial.
- Detallar el proceso entrenamiento de la red neuronal.
- Evaluar el sistema propuesto para verificar su funcionamiento.
- Determinar la mejor forma para alertar la falta de uso de uno o varios equipos de protección personal de los técnicos.

1.5 Alcance del proyecto

1.5.1 Delimitación física

El desarrollo del trabajo de investigación toma como ubicación geográfica el laboratorio clínico “VIDASSAN”, localizado en José María Egas Cdla. Alborada 7ma Etapa, Mz. 738 – villa 11, en la urbe guayaquileña.

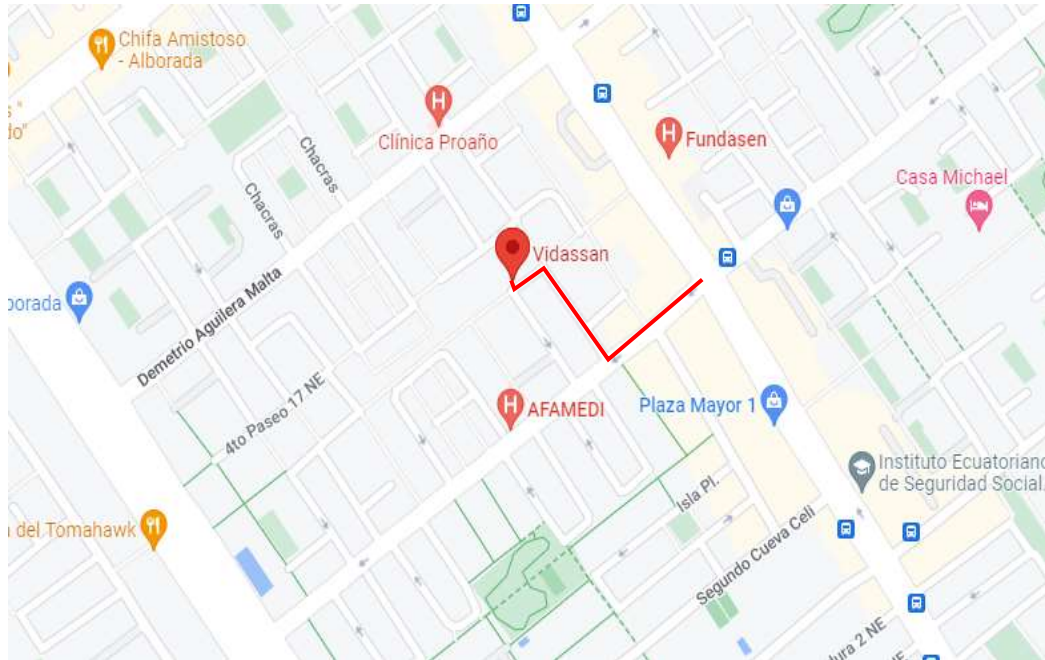


Figura 1. Ubicación geográfica del laboratorio clínico “VIDASSAN”, 2022. Información tomada de Google Maps. Elaborado por el autor

1.5.2 Alcance

Desarrollo de un sistema de detección de equipo de seguridad de los técnicos de un laboratorio mediante el procesamiento de vídeo en tiempo real a través de la cámara integrada en una laptop para el reconocimiento de utilizar los equipos de protección personal más importantes tales como los guantes de nitrilo y la mascarilla quirúrgica para la realización de su trabajo bajo medidas de bioseguridad.

1.6 Metodología

El presente proyecto está adaptado la metodología de tipo cualitativa e incremental del desarrollo de software, a través de cuatro etapas: elección de herramientas, codificación, entrenamiento y pruebas del sistema con el objetivo de tener

retroalimentación del funcionamiento del sistema y así conocer si existen cambios necesarios, mediante cuatro incrementos.

El incremento uno aportó con un análisis de las herramientas de programación, así como los sistemas de detección de videos ya usados, para establecer cuál era el más óptimo para la elaboración del sistema propuesto, instalando las librerías necesarias juntos con los softwares.

En el incremento dos se entrenó la red neuronal sólo para el reconocimiento del equipo de seguridad personal en esta especialidad, tales como mascarilla y guantes de Nitrilo usados dentro del laboratorio, por lo que el sistema no es aplicable para el reconocimiento de cualquier objeto.

De tal manera con el incremento tres se tomaron fotografías para establecer un banco de imágenes de los equipos de protección personal que se usan los técnicos de laboratorio para entrenar la red neuronal lo que permitió conseguir resultados más efectivos del reconocimiento de dichos objetos mediante la visión artificial.

Como último, en el incremento cuatro se realizaron evaluaciones del funcionamiento correcto del sistema para en un futuro la implementación dentro del laboratorio clínico.

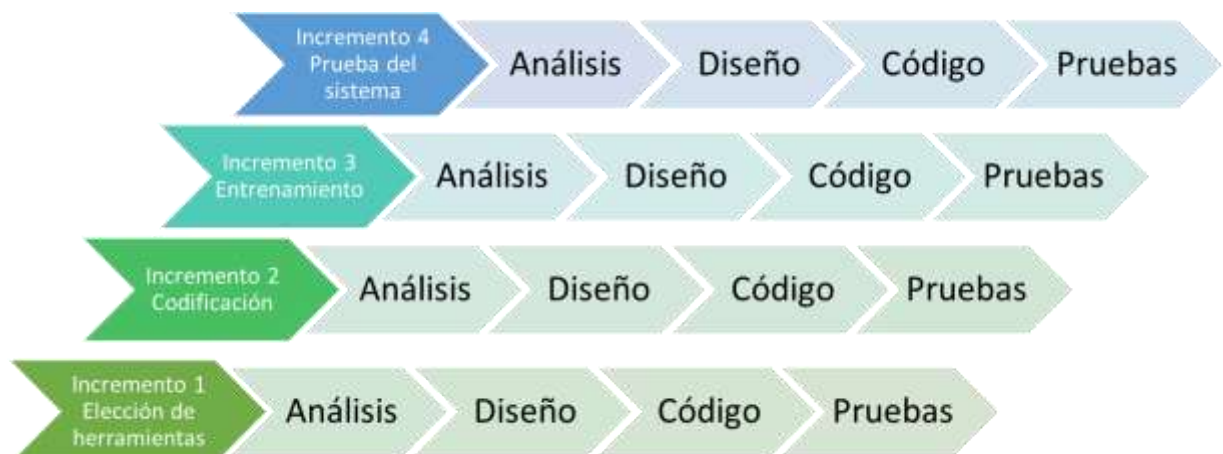


Figura 2. Metodología del desarrollo de software incremental, 2022. Elaborado por autor.

CAPÍTULO II

MARCO TEÓRICO

2.1 Riesgos en un laboratorio

2.1.1 Riesgo

Los peligros son los efectos dañinos descritos de forma cualitativa, sin embargo, el riesgo es una manera cuantitativa para medir cuán probable es que estos sucedan. Al ser el riesgo medible se calcula su magnitud con la siguiente fórmula $\text{magnitud} = \text{probabilidad} \times \text{consecuencias}$.

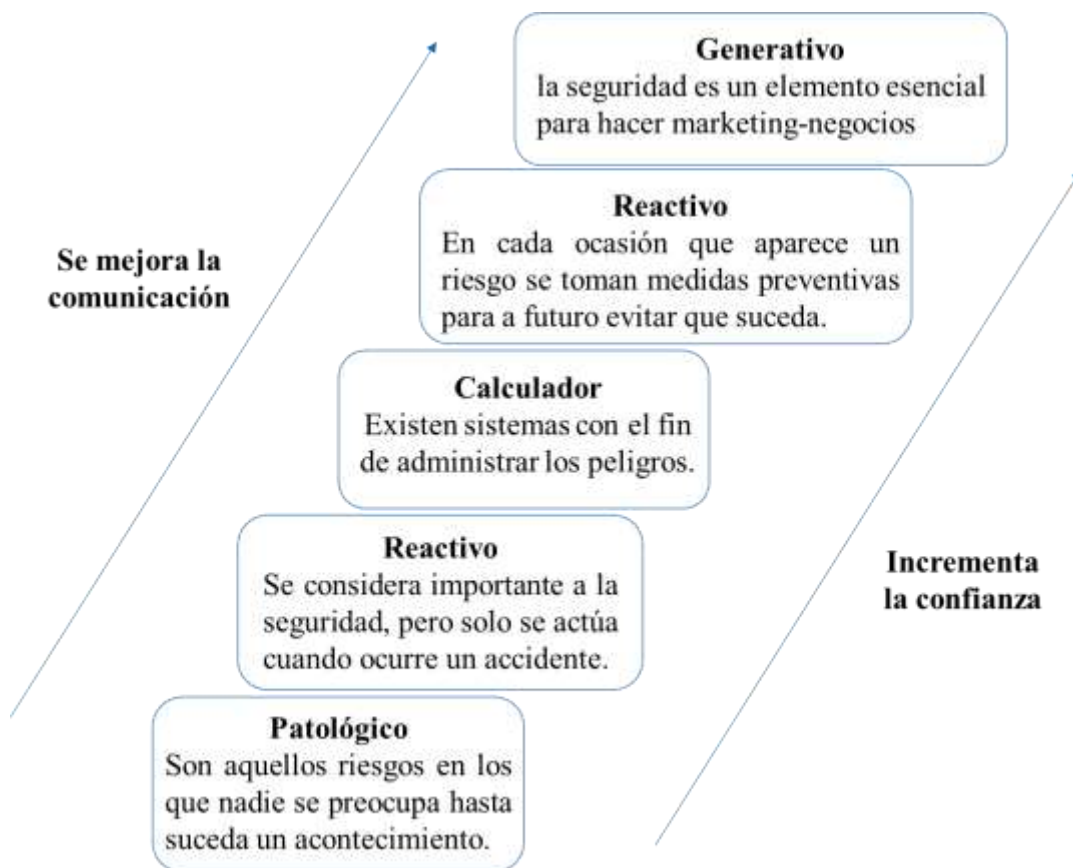


Figura 3. Evolución de la cultura de seguridad, 2008. Información adaptada de la Revista del Hospital Materno Infantil Ramón. Elaborado por Susana Der Parsehian.

2.1.2 Tipos de peligros en un laboratorio

Un paso importante para salvaguardar la salud, y proteger la integridad física y biológico de los técnicos o colaboradores del laboratorio es reconocer los peligros en el lugar de trabajo. Es posible categorizar al peligro en tres tipos que son: químicos, biológicos o físicos (Weng, 2005).

2.1.2.1 Peligros químicos

Un químico peligroso puede ser un sólido, líquido o gas. Puede ser una sustancia pura, que consta de un ingrediente, o una mezcla de sustancias. Puede dañar la salud de una persona que está expuesta a ella.

El efecto nocivo que se puede producir en la salud del trabajador es por contacto directo o al estar expuestos a la sustancia química, también se puede producir cuando se inhalan estos químicos, cuando hay contacto directo en la piel o cuando se ingieren. El uso de productos químicos en los laboratorios de investigación es inevitable, y existe una gran posibilidad de que se generen daños o lesiones podría ser importante si se utilizan o manipulan de forma indebida (Beltrón, 2020).



Figura 4. Símbolos de productos químicos, 2016. Información tomada de la Universidad de “les Illes Balears”. Elaborado por Prevencio.

2.1.2.2 Peligros biológicos

El investigador Beltrón (2020) menciona que los peligros biológicos abarcan microbios, organismos recombinantes y vectores virales. Es una tarea común que los técnicos de laboratorio trabajen con sustancias biológicas, la más común es la sangre, pero hay más tipos que representan riesgos para ellos, es entonces se demuestra la prioridad de que el empleador le garantice seguridad al laborar y los procesos se puedan realizar de forma infalible.

Los riesgos biológicos son una preocupación en los laboratorios en los que se manipulan microorganismos o material contaminado con ellos. Estos peligros suelen estar presentes en los laboratorios de investigación de enfermedades infecciosas y clínicas, donde se manipulen fluidos corporales, tejidos o líneas celulares primarias o inmortalizadas de origen humano o animal (Pence, 2018). Además, en donde se utilicen microorganismos, incluidos los vectores virales de replicación deficiente, para la expresión de proteínas u otros métodos in vitro. Ocasionalmente, los riesgos biológicos están presentes en los laboratorios de prueba y control de calidad.



Figura 5. Riesgo biológico de usar sangre en un laboratorio, 2020. Información tomada de iStock, Elaborado por FabrikaCR,

2.1.2.3 Peligros físicos

Las instalaciones de investigación inherentemente tienen peligros físicos significativos presentes. Aquí se incluyen los riesgos de seguridad eléctrica, los riesgos ergonómicos relacionados al uso manual de materiales como de equipos, además, emplear objetos punzocortantes y cuestiones básicas de limpieza.

Muchas operaciones en el laboratorio pueden resultar en que los trabajadores del laboratorio asuman posturas incómodas repetitivas o sostenidas. El trabajo durante períodos prolongados en un gabinete de bioseguridad o la observación de portaobjetos en un microscopio durante períodos prolongados genera malestar en la espalda. Es posible provocar un gran problema en la ergonomía de los colaboradores cuando realiza una actividad en una misma posición durante un tiempo prolongado, un resultado precipitado es que la persona siente dolor en la parte de su cuerpo que ha estado más forzado, por lo que se recomienda que se inculque colocarse en una postura equilibrada y que no se contraigan tantos músculos. Las pipetas programables o de asistencia magnética pueden reducir la frecuencia de la fuerza manual necesaria para evitar lesiones en los trabajadores.

Los contenedores de objetos punzantes son omnipresentes en los laboratorios de investigación y seguir algunas reglas de seguridad puede ayudar a evitar quedarse con los informes de accidentes. Muchas lesiones se derivan de una limpieza deficiente (Islas Baleras, 2016).

Existen peligros físicos que son evitables pero muy comunes tales como resbalones, tropezones o caídas. Es importante evitar que el almacenamiento de los residuos sean un peligro, algunos ejemplos son las fundas de basura, los contenedores, etc., puesto que se debe evitar que las agujas residuales puedan producir heridas, otro punto importante es la cantidad de contenedores o fundas que se apilan porque pueden deslizarse o colapsarse. Es por eso que se deben de mantener el espacio del laboratorio libre, sin obstáculos para prevenir peligros como los antes mencionados, y otros más como ser refugio de plagas.

2.2 Vías de transmisión de infecciones dentro del laboratorio

Las vías de transmisión a las que los técnicos del laboratorio están expuestos se pueden dividir en tres vías a través: del sistema respiratorio, de la piel, por el sistema digestivo como se demuestra a continuación.

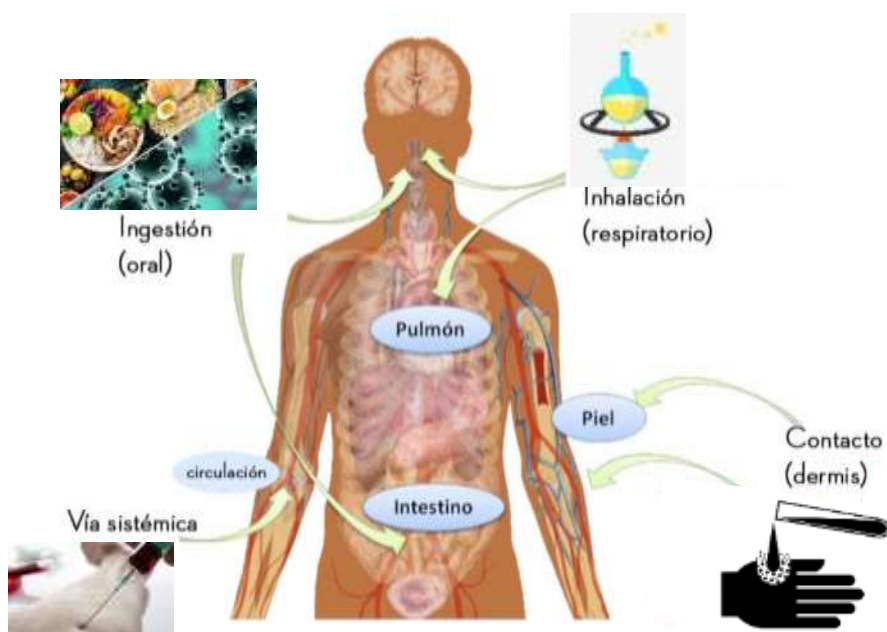


Figura 6. Vías de transmisión de infecciones, 2017. Información adaptada de la revista de Educación Bioquímica. Elaborado por Elda Coutiño, Lucerito Lagunes, Omar Helguera.

2.2.1.1 A través del aparato respiratorio

Este riesgo se produce al inhalar aerosoles y gotas, los cuales son en su mayoría producidos de forma involuntaria ya que los colaboradores dentro del laboratorio miden con las pipetas, vierten líquidos, y a la vez los agitan generando salpicaduras, incluso se realizan procesos en el que se evaporan los líquidos generando gases inflamables para el organismo, se identifica también que al realizar pruebas de catalasa se producen partículas en el aire las mismas que no sobrepasan los 0.05 milímetros de diámetro.

Como se especificó antes cuando se realizan procesos químicos se generan partículas en ocasiones no son tan pequeñas, algunas logran superar los 0.1 milímetros de diámetro las mismas que pueden infectar el aire para que posteriormente contaminen los guantes que los colaboradores usan, y las superficies de trabajo dentro del laboratorio.

2.2.1.2 A través de la piel o mucosas

Se refiere cuando se introducen materiales infecciosos con objetos punzocortantes contaminados tales como agujas, cuchillas o algún tipo de cristal roto.

Derrames directos a la piel, salpicadura o rociadas que pueden llegar a los ojos, boca y nariz, e incluso cuando se coloca la mano en la cara después de haber trabajado con químicos son ejemplos de cómo se provoca contaminación tópica. Además, cuando uno de los técnicos tiene abrasiones, o cortes superficiales en la piel también son otro medio de contaminación dentro del laboratorio.

2.2.1.3 A través del tubo digestivo

Al aspirar con una pipeta, llevarse lápices, esferos o los dedos en la boca o cerca de esta, consumir alimentos o bebidas dentro del laboratorio, o cuando se producen salpicaduras accidentales en la boca.

2.2.2 Infecciones adquiridas en el laboratorio

Estar dentro de un laboratorio representa un alto riesgo para cada uno de los colaboradores que trabajan en el mismo, es por esto que a continuación se detallarán las enfermedades más comunes de infectarse, las cuales son:

2.2.2.1 Gripe

Es una enfermedad respiratoria que se transmite muy fácilmente, principalmente afecta la nariz, garganta e incluso hasta los pulmones. Paulatinamente Es posible que se convierta en una enfermedad grave capaz de provocar la muerte.

2.2.2.2 Varicela

Los investigadores Xavier y Erazo (2022), describen a la varicela como la enfermedad provocada por el patógeno de la varicela-zóster, este genera ampollas, comezón, fatiga y fiebre

2.2.2.3 Tuberculosis

La tuberculosis es una enfermedad peligrosa porque cuando una persona infectada tose transmite el virus a las personas cercanas a él, principalmente perjudica a los pulmones, pero también puede llegar a afectar el cerebro y la columna vertebral.

2.2.2.4 Salmonelosis

Es una enfermedad bacteriana que afecta directamente al intestino cuando se consumido algún alimento infectado por virus provenientes de los desperdicios biológicos de los animales.

2.2.2.5 Hepatitis A

La hepatitis A es una infección que genera daño en el hígado, el cual se transmite como consecuencia del contacto con desperdicios humanos de una persona ya contagiada.

2.2.2.6 Hepatitis B

La hepatitis B se trasmite cuando una persona tiene contacto directo con saliva, sangre, semen, o con un objeto contaminado, como un cepillo de dientes o una navaja, donde el virus puede vivir durante días.

2.2.2.7 Hepatitis C

Una persona se puede contagiar con la hepatitis C cuanto esta tiene contacto con sangre contaminada. La hepatitis C es contagiosa y puede causar daño hepático grave incluso si la persona nunca presenta síntomas.

2.2.2.8 VIH

Sus siglas significan virus de la inmunodeficiencia humana, se contagia por el contacto que se puede tener con sangre infectada. Se refiere a un virus que afecta a las células encargadas de defendernos de las infecciones, debilitando el sistema inmune, y como consecuencia la persona que lo padece es mucho más vulnerable y sensible ante otras enfermedades e infecciones; lo que provoca que la persona sea más vulnerable a otras infecciones y enfermedades. (Coronel, 2020)

2.2.2.9 Tétano

El tétano es una enfermedad muy dañina que lastima o afecta directamente el sistema nervioso central debido a que la bacteria produce toxinas, lo que la hace una enfermedad incurable.

2.2.2.10 COVID-19

Enfermedad que el 2020 fue declarada a nivel mundial como una pandemia porque su transmisión es muy fácil, la misma se produce por un tipo de virus respiratorio del segundo tipo (SARD-CoV-2), afecta a muchos órganos, dentro de sus síntomas tenemos fiebre, problemas de respiración, las papilas gustativas no pueden detectar sabores, produce una fuerte tos, por lo que puede generar un cuadro agudo en los pacientes hasta la muerte, tiene una tasa de mortalidad del 1% al 3%. (Francisco Díaz, 2020)

2.2.3 Seguridad laboral

2.2.3.1 Equipos de protección personal

Los EPP (equipos de protección personal) considerados como indumentarias de protección que comprenden un conjunto de barreras para evitar que los trabajadores entren en contacto con agentes infecciosos o que deterioran la salud. El aislamiento o la eliminación de la fuente de riesgo debe ser la primera acción preventiva. (Romero, Mesa, Estrella, & Ramírez, 2021)

La decisión de qué y dónde utilizar los EPP está influida tanto por la percepción del riesgo como por los factores organizativos cognitivos y físicos de la actividad. La prevención de la propagación de organismos infecciosos no sólo debe depender del uso de los EPP, sino que también debe reconocerse como una responsabilidad organizativa

2.2.3.2 Bata de laboratorio

Se recomiendan batas o uniformes protectores de laboratorio para evitar la contaminación de la ropa personal. quitarse la ropa protectora antes de salir a áreas que no sean del laboratorio. Se debe desechar la ropa de protección de un solo uso con otros desechos contaminados o depositar la ropa reutilizable para su lavado por parte de la institución.

2.2.3.3 Protección de mano

Se recomienda usar guantes desechables mientras se trabaja con agentes que puedan corroer la piel, ya que el personal es variado no solo en género, sino también por su tamaño, se recomienda entonces que se tenga a disponibilidad de los colaboradores diversos tamaños de guantes, para que ellos puedan escoger el que mejor se ajusta. Antes de comprarlos, se debe verificar la información del fabricante de que estos cumplan los estándares de pruebas y así corroborar que soporten los peligros a los que se pueden exponer, la gerencia del laboratorio o un oficial de seguridad son los encargados.



Figura 7. Guantes de nitrilo, 2020. Información tomada de NOVASAN

Dentro del mercado existen muchos tipos de guantes, los mismo que se clasifican dependiendo de su composición, en la siguiente tabla se describen las características principales las que se toman en consideración al momento de escoger el tipo de guante a usar dentro del laboratorio, como se muestra a continuación:

Tabla 1. *Tipos de guantes según su composición*

Tipos de guantes	Descripción
Natural Material: Látex que es un caucho natural	Aseguran ser efectivos, y generar comodidad, también la sensibilidad al tacto porque se ajusta bien en la mano. Se consideran como la primera opción por su asequibilidad.
Sintético Material: PVC, nitrilo, vinilo, neopreno, elastireno, tactilón, poliisopropeno.	Estos guantes son más resistentes y aseguran una mejor protección ante microorganismos y agentes químicos. Son también otra alternativa para evitar alergias por el látex
Guantes tricapa	Como su nombre lo indica tienen 3 capas, la externa es de látex, la capa media y la capa que toca a la piel son de nitrilo que se considera como un material sintético

Información adaptada del repositorio virtual de la universidad de Barcelona, 2018. Elaborado por Martín Ambrona y Antonio Jesús

2.2.3.4 Protección para las vistas y el rostro

Es recomendable también que para las vistas y el rostro se usen equipos de protección, tales como mascarillas y gafas siempre que pueda ocurrir un evento de salpicadura o rociado. Esto incluye abrir contenedores y pipetear, manipular, dividir en alícuotas o analizar muestras, cultivos, agentes biológicos u otros materiales peligrosos.

Existen muchos tipos de mascarillas, pero cada de estas filtran las partículas en el aire en diferente porcentaje como se muestra a continuación:



Figura 8. *Características de las mascarillas quirúrgicas, 2021, Información tomada de Seguro Social de Salud del Perú*

Tabla 2. Descripción de los tipos de mascarillas

Tipo	Descripción
Mascarillas quirúrgicas	Se filtra con un porcentaje mínimo del 80% de las partículas en suspensión, se daña con el contacto de aceite.
N95	Se filtra con un porcentaje mínimo del 95% de las partículas en suspensión. Se daña con el contacto de aceite.
N95 quirúrgica	Se filtra con un porcentaje mínimo del 95% de las partículas en suspensión, conservan las partículas que absorbe, pero se daña con el contacto de aceite,
N99	Se filtra con un porcentaje mínimo del 99% de las partículas en suspensión. Se daña con el contacto de aceite.
N100	Se filtra con un porcentaje mínimo del 99,97% de las partículas en suspensión. Se daña con el contacto de aceite.
R95	Se filtra con un porcentaje mínimo del 95% de las partículas en suspensión. Es un poco resistente al aceite.
P95	Se filtra con un porcentaje mínimo del 95% de las partículas en suspensión. Resiste al contacto con óleo.
P99	Se filtra con un porcentaje mínimo del 99% de las partículas en suspensión. Resiste al contacto con óleo.
P100	Se filtra con un porcentaje mínimo del 99,97% de las partículas en suspensión. Resiste al contacto con óleo.

Adaptado del Journal of Healthcare Quality Research, 2020, Elaborado por Andrés, de Castro, Vicente-Guijarro, Peribáñez, J., Haro, M., Valencia-Martín, J., ... & Montero, G., sciencedirect

2.2.4 Visión artificial

Se busca aplicar el mismo procedimiento que se realiza una persona para identificar un objeto en una imagen, es por esto que se desarrollan sistemas de manera digital con la capacidad de procesar, realizar análisis, y comprender los datos que se obtienen de forma visual a través de imágenes o vídeos. Para que una computadora

tenga esa capacidad es necesario que se la entrene. En particular lo que se busca es que los equipos sustraigan datos visuales, aprenda a generar resultados cuando se le aplique un algoritmo y detecte.

Las aplicaciones de visión artificial se utilizan en una variedad de industrias para realizar diversas tareas, incluidas las siguientes:

- **Análisis de componentes electrónicos.** La visión artificial se utiliza en la construcción de placas de circuitos para tareas como la inspección de soldadura en pasta y la colocación de componentes.
- **Reconocimiento óptico de caracteres:** permite que una computadora extraiga texto impreso o escrito a mano de las imágenes.
- **Reconocimiento de escritura y firma:** Con estas características, una computadora puede detectar patrones en imágenes de escritura a mano y firmas.
- **Reconocimiento de objetos:** En la industria automotriz, los automóviles autónomos utilizan el reconocimiento de objetos en imágenes tomadas por cámaras para identificar obstáculos en la carretera. Los sistemas de visión artificial también determinan la posición de los objetos, como la colocación adecuada de una etiqueta en un frasco de pastillas.
- **Reconocimiento de patrones:** utiliza el reconocimiento de patrones para realizar diagnósticos basados en tecnologías como la resonancia magnética, los análisis de sangre y los escáneres cerebrales.
- **Inspección de materiales:** Las capacidades de visión artificial en los sistemas de inspección de materiales garantizan el control de calidad. La visión artificial verifica fallas, defectos y contaminantes en una variedad de materiales y productos. Por ejemplo, estos sistemas pueden inspeccionar píldoras y tabletas en busca de problemas durante la fabricación.
- **Inspección de moneda:** La visión artificial se utiliza para analizar monedas y detectar billetes falsos.
- **Conteo de artículos:** Esta capacidad se utiliza para contar elementos como pastillas en un paquete o botellas en un estuche.

En conclusión, las computadoras analizan las imágenes como diferentes recuadros o también llamados pixeles, cada pixel de forma individual contiene un grupo

de valoración de color. En la imagen 9 se toma como ejemplo al expresidente de Estados Unidos, Abraham Lincoln, esta imagen posee diferentes matices unos con más claridad y otros más oscuros, por lo que la representación del brillo en cada pixel, se le ha otorgado un valor de 8 bits, al negro siendo este el que no posee brillo un valor de 0, y el punto o pixel más blanco el valor de 255, de esta manera es como la computadora lee una imagen, y se los toma como el comienzo para su algoritmo, logrando procesar y analizar, para que sea capaz de decidir. En la figura 9, se muestra como los valores de color de los píxeles individuales se convierten en una matriz simple de números utilizados como entrada para un algoritmo de visión por computadora.

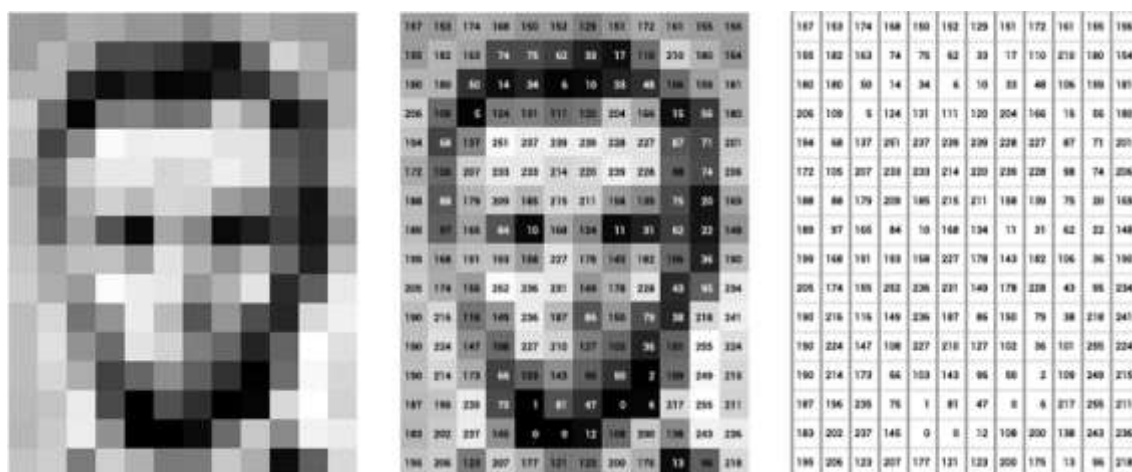


Figura 9. Proceso de imagen para visión artificial, 2020. Información tomada de Adobe, Elaborado por Golan Levin

2.2.4.1 Visión artificial en IA

La IA se utiliza en la visión artificial para determinar la mejor interpretación. La IA puede procesar una amplia cantidad de imágenes e información de datos que antes era demasiado difícil de recopilar.

Ejemplos de cómo se utiliza la IA con la visión artificial son los siguientes:

- El reconocimiento de firmas y caracteres requiere un nivel de matiz con el que la IA puede ayudar.
- En la fabricación, la IA ayuda con el reconocimiento de objetos y la inspección de materiales para permitir que los sistemas de visión artificial comprendan las variaciones aceptables en la forma y la textura de un objeto o material.

- En el control de calidad, un sistema habilitado para IA puede interpretar anomalías aceptables en lugar de rechazar cualquier cosa que no se ajuste estrictamente a una especificación.

2.2.5 Aprendizaje automático

Los algoritmos de aprendizaje automático son algoritmos matemáticos que permiten a las máquinas aprender imitando la forma en que aprenden los humanos, aunque el aprendizaje automático no son sólo algoritmos, sino también el enfoque desde el que se aborda el problema. El aprendizaje automático es esencialmente una forma de obtener inteligencia artificial. Muchos problemas de diferentes áreas han sido resueltos con la ayuda del aprendizaje automático. A diferencia de unos años atrás cuando se deseaba crear sistemas o programas este requería una gran cantidad de tiempo y de recursos en especial recurso humano, para poder programar su funcionamiento, muy por lo contrario de la cantidad de recursos que ahora se necesitan gracias al aprendizaje automático. Desde hace un tiempo, al tener disponibles grandes fuentes de datos y la viabilidad de los recursos informáticos, el aprendizaje automático se ha convertido en algo predominante con una amplia gama de aplicaciones en nuestra vida cotidiana. Se dividen en tres tipos de aprendizaje que son:

- Supervisado
- No supervisado
- Por refuerzo

2.2.5.1 Aprendizaje supervisado

Es el tipo de aprendizaje más elemental de algoritmos de aprendizaje automático. Este tipo de aprendizaje exige una supervisión directa. En donde es necesario ingresar información y datos de los objetos que previamente fueron etiquetados por los humanos, en los cuales indican el objeto o los objetos de interés, para que a través de un algoritmo el computador "aprenda". Al final, el algoritmo aprende de los datos anotados y predice las anotaciones de los nuevos datos que el algoritmo no conocía hasta ese momento, una vez completado el proceso de entrenamiento. Dentro de esta clasificación existen algunos algoritmos que son los más usados tales como:

- Redes neuronales
- Árboles de decisión

- Bosque aleatorio
- K-Nearest Neighbors (vecinos más cercanos)
- Regresión lineal
- Regresión logística
- Máquinas de vectores de apoyo

2.2.5.2 Aprendizaje no supervisado

Este tipo de algoritmo intenta aprender e identificar propiedades útiles de las clases a partir de los datos anotados, sin la ayuda o la intervención de un humano. Algunos de los tipos de aprendizaje no supervisado más conocidos son el algoritmo Apriori, agrupamiento tipo k-means, entre otros

2.2.5.3 Aprendizaje por refuerzo

Este aprendizaje permite a la máquina entrenarse continuamente mediante el método de ensayo y error. Como resultado, la máquina estudia el error o la experiencia fallida anteriormente por lo que procura obtener de estos datos que le permitan mejorar en la próxima predicción, y así ser más preciso. El proceso de decisión de Markov, el aprendizaje Q, la diferencia temporal, etc. son algunos de los ejemplos de aprendizaje por refuerzo.

2.2.6 Redes neuronales artificiales

Conocidas como el tipo más común de la clasificación del aprendizaje supervisado. El nombre a este modelo se lo dio porque se desarrollaron para imitar al cerebro humano y la función neuronal que genera y con la que funciona. Una red neuronal artificial consiste tener conectadas neuronas interrelacionadas, las mismas que se trabajan entre ellas en capas para replicar la función neuronal de nuestro cerebro. Se busca imitar la estructura y funcionamiento del cerebro humano, por lo que cada neurona de este tipo de entrenamiento realiza unidades de cálculo.

Cada conexión que se realiza de una neurona a otra es conocido como sinapsis y esta contiene datos numéricos. Por lo tanto, en un sentido simple, cuando se proporciona un valor de entrada en una neurona (x_1, x_2, \dots, x_n), éste atraviesa la sinapsis, multiplicando su valor con el valor ponderado de la sinapsis (w_1, w_2, \dots, w_n), se añade entonces a la suma de estos valores. Esta será la salida de la neurona. Dado que una neurona no conoce su límite, se requiere un mecanismo de mapeo para asignar las entradas a la salida, conocido como "función de activación".

En una red multicapa totalmente conectada, todas las salidas de una capa de neuronas se convierten en la información entrante a todas las neuronas de las capas posteriores. Como resultado, algunas capas procesan los datos de entrada originales, mientras que otras procesan los datos obtenidos de las neuronas de la capa anterior. Por lo tanto, el número de pesos de cualquier neurona de la red es el mismo que el de la capa previa a la de la neurona en cuestión.

Las redes neuronales artificiales constan de tres características fundamentales: la arquitectura de la red, las funciones de entrada y activación y el peso de las conexiones de entrada. La arquitectura y las funciones de la red se eligen en la fase inicial y se mantienen igual durante el entrenamiento. Se considera directamente proporcional el valor de los pesos para el rendimiento de la red neuronal. Los pesos se ajustan durante el entrenamiento para conseguir un determinado resultado. Las RNA pueden entrenarse utilizando una multitud de programas de entrenamiento diferentes.

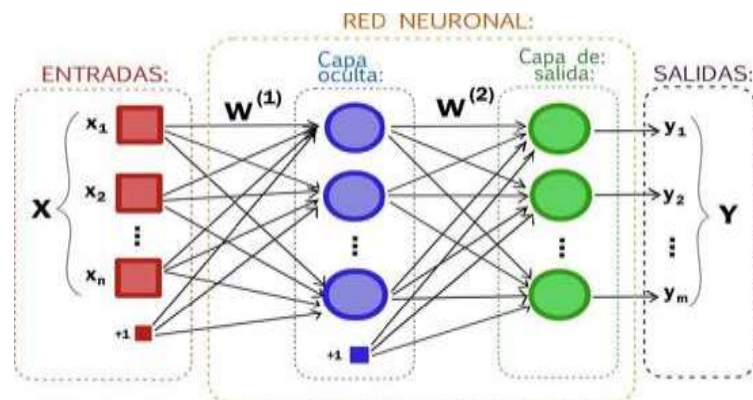


Figura 10. Estructura de una red neuronal, 2019. Información tomada de la Revista CEO, Elaborado por Juan Reyes, Magda Sánchez, Raúl Mejía.

2.2.7 Redes neuronales convolucionales

Las redes neuronales convolucionales han adquirido un estatus especial en los últimos años como una forma especialmente prometedora de aprendizaje profundo. Arraigadas en el procesamiento de imágenes, las capas convolucionales se han abierto camino en prácticamente todos los subcampos del aprendizaje profundo, y son muy exitosas en su mayor parte. La diferencia fundamental entre las redes neuronales totalmente conectadas y las convolucionales es el patrón de conexiones entre capas consecutivas. En el caso de las redes totalmente conectadas, como su nombre indica, cada nodo está enlazado con cada unidad de la capa previa. Por ejemplo 10 unidades de salida estaban conectadas totalmente conectadas a cada pixel de la imagen que ingresa.

Además, todas las unidades se conectan a la capa anterior de la misma manera, con los mismos pesos y estructura. Esto conduce a una operación conocida como convolución, que da nombre a la arquitectura.

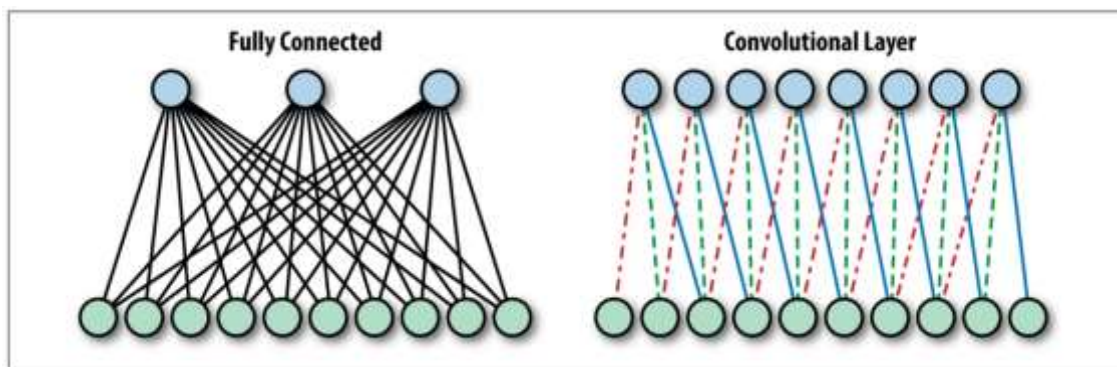


Figura 11. Estructura de red completamente conectada y convolucional, 2017. Información tomada del artículo *Learning Tensorflow*, Elaborado por Hope T., Yehezkel S. & Lieder I.

Las redes neuronales siguen el patrón, cada capa convolucional examina una parte cada vez mayor de la imagen a medida que se profundiza en la red. Lo más habitual es que a esto le sigan capas totalmente conectadas que, en la analogía de inspiración biológica, actúan como los niveles superiores de procesamiento visual que se ocupan de la información global.

El segundo punto de vista, más orientado a la ingeniería de datos, se deriva de la naturaleza de las imágenes y su contenido. Cuando buscamos un objeto en una imagen, por ejemplo, la cara de un gato, normalmente queremos poder detectarlo independientemente de su posición en la imagen. Esto refleja la propiedad de las imágenes naturales de que el mismo contenido puede encontrarse en diferentes lugares de una imagen. Esta propiedad se conoce como invariancia, se pueden esperar invariancias de este tipo con respecto a (pequeñas) rotaciones, cambios en las condiciones de iluminación, etc. En consecuencia, cuando se construye un sistema de reconocimiento de objetos, éste debe ser invariante a la traslación y, dependiendo del escenario, probablemente también a la rotación y a las deformaciones de muchos tipos.

En pocas palabras, tiene sentido realizar el mismo cálculo exacto en diferentes partes de la imagen. Desde este punto de vista, una capa de red neuronal convolucional calcula las mismas características de una imagen, en todas las áreas espaciales. Por último, la estructura convolucional puede verse como un mecanismo de regularización. Desde este punto de vista, las capas convolucionales son como las capas totalmente conectadas, pero en lugar de buscar los pesos en todo el espacio de matrices (de cierto tamaño), se limita la búsqueda a las

matrices que describen convoluciones de tamaño fijo, reduciendo el número de grados de libertad al tamaño de la convolución, que suele ser muy pequeño.

2.2.7.1 Arquitectura de las redes convolucionales

La arquitectura de redes convolucionales se compone de 5 capas, cuando estas se apilan, se formará una arquitectura RNC. Por otra parte, de las capas que conforman esta arquitectura, hay dos factores considerables que son, la capa de abandono y la función de activación.

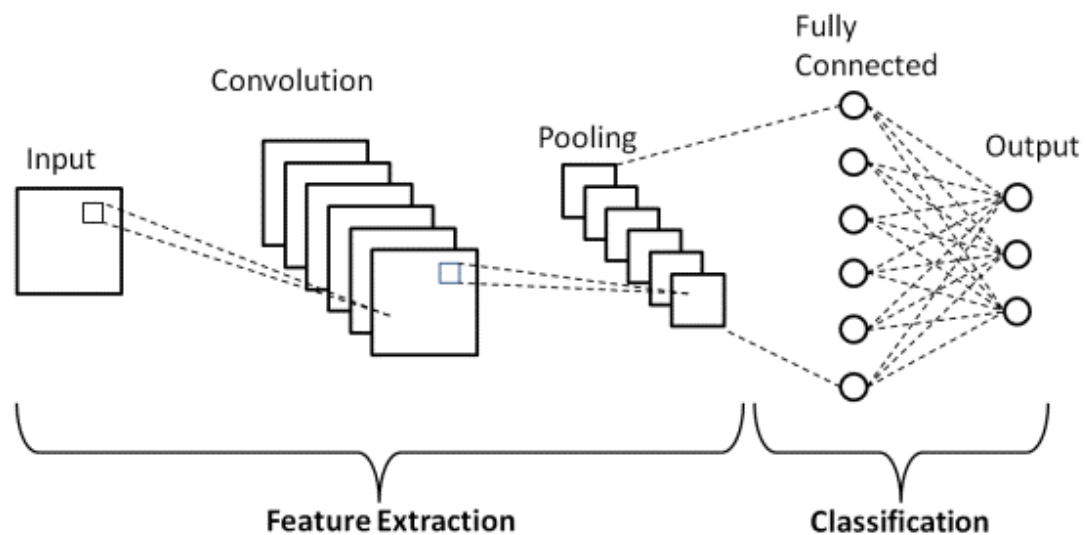


Figura 12. Arquitectura de las redes convolucionales, 2020, Información tomada de Techeopedia. Elaborado por Sai Balaji

Capa Convolucional: Esta capa es la primera capa que se utiliza para extraer las diversas características de las imágenes de entrada.

Aquí se realizan todos los procesos numéricos de convolución, en medio de los datos de la imagen que ingresa, y la aplicación de un filtro en específico resulta en un matriz de datos con las porciones de las imágenes y del tamaño según lo indicado.

A estas salidas se las conoce como los mapas de características, donde se indica información de la imagen. Para después con estos resultados alimentar capas para que con el proceso aprendan las características de la imagen que se ingresan.

El resultado de la capa de convolución en RNC, se envía después a la capa posterior, siempre y cuando se hayan aplicado una convolución al inicio. Las RNC mejoran su rendimiento cuando se aplica este filtro y se fija que la posición en el espacio de los píxeles de la imagen esté íntegra. La capa de convolución en RNC pasa el resultado a la siguiente

capa una vez que se aplica la operación de convolución en la entrada. Las capas convolucionales en RNC se benefician mucho ya que asegura que la relación espacial entre los píxeles estén intactos.

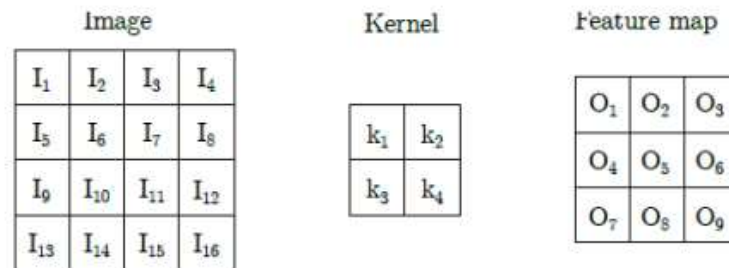


Figura 13. Arquitectura de las redes convolucionales, 2018. Información adaptada de Quora. Elaborado por Rahman N

Capa de agrupación: Frecuentemente, este tipo de capa va frente a la capa de convolución. Cuya meta es reducir el tamaño del mapa de características en el que ya se hizo convolución, lo que genera usar menos memoria RAM, y que el uso de la unidad de procesamiento gráfico sea insignificante, porque se quitan algunos enlaces entre las capas. Existen varios métodos, y para cada uno una forma de agrupar las características que vienen de la capa de convolución.

En Max Pooling, el componente más considerable es del que proviene del mapa de características. En un tamaño definido la agrupación promedio se calcula ponderando cada parte de una imagen dentro de este rango. Una vez que se tengan todas las divisiones se pueden sumar en la sección predeterminado se lo puede realizar en Sum Pooling. El puente entre las capas previas, que son la convolucional y de conexión completa es la capa de agrupación. Este modelo RNC generaliza las características extraídas por la capa de convolución y ayuda a las redes a reconocer las características de forma independiente. Lo que permite también que se reduzcan los cálculos matemáticos en una red.

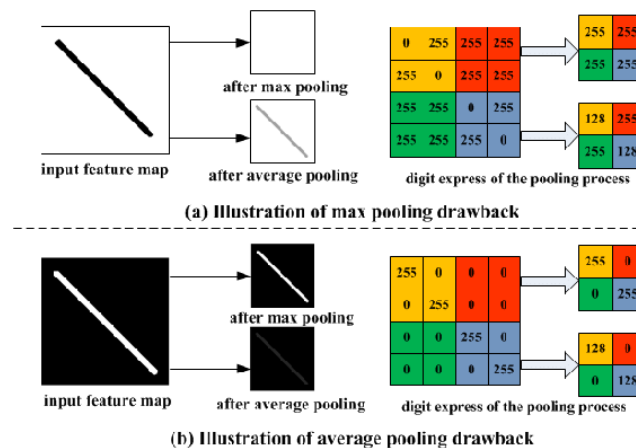


Figura 14. Proceso de max pooling, 2018. Información adaptada de Quora. Elaborado por Rahman N.

Capa completamente conectada: La capa totalmente conectada (TC) consta de los pesos y sesgos junto con las neuronas y se utiliza para conectar las neuronas entre dos capas diferentes. Mayormente se las coloca previo a la última capa de salida. Y también son parte de las últimas capas de RNC.

La imagen que ingresa es sometida a un aplanamiento en las capas anteriores, y sus resultados sirven para la capa totalmente conectada. Posterior a esto nuevamente pasa por capas de aplanamiento y en estas se aplican el algoritmo matemático para que vaya aprendiendo, y es aquí en donde se empieza a clasificar. Se argumenta entonces que el uso de dos capas conectadas en su totalidad generará buenos resultados a comparación de que la RNC solo tenga una. Esto permite que la RNC no requiera de supervisión humana,

Abandono: Al tener enlazados todos los detalles a la capa, existe la posibilidad de ocasionar un sobreajuste, completo o parcial en el dataset de las imágenes de entrenamiento. Ocurre el sobreajuste cuando un gran número de entrenamiento, funciona muy bien pero solo con elementos que conoce, cuando se le aplica la carpeta de test arroja error porque desconoce esos objetos. Para superar este problema, se utiliza una capa de eliminación en la que se borran algunas neuronas de la red neuronal durante el proceso de entrenamiento, lo que reduce el tamaño del modelo. Cuando se abandonan nodos se los hace de forma aleatoria.

La deserción mejora el rendimiento de un modelo de aprendizaje automático, ya que evita el sobreajuste al simplificar la red. Suelta neuronas de las redes neuronales durante el entrenamiento.

Funciones de activación: Estas funciones son las que se utilizan para aprender y semejar alguna relación continua y compleja que encuentren entre las características de las variantes de la red. En palabras simples, decide qué información del modelo debe dispararse hacia adelante y cuál no al final de la red.

Agrega no linealidad a la red, hay varias funciones de activación de uso común, como las funciones ReLU, Softmax, tanH y Sigmoid. Cada función tiene como objetivo cumplir un uso especial. Para un modelo RNC de clasificación binaria, se prefieren las funciones sigmoide y softmax y para una clasificación multiclase, generalmente se usa softmax. En general, la función de activación funciona como un switch en donde solo va a

activar o desactivar según lo considere, decide también si los datos que ingresan son importantes o no para procesarlos de manera numérica.

En términos simples, las funciones de activación en un modelo RNC determinan si una neurona debe activarse o no. Decide si la entrada al trabajo es importante o no para predecir usando operaciones matemáticas.

2.2.8 Aprendizaje profundo

El aprendizaje profundo se denomina como una técnica usada para sustraer datos y posteriormente transformarlos, este proceso lo realiza mediante capas de redes neuronales lo que ha hecho posible que se pueda reconocer la voz humana, e incluso clasificar animales a través de imágenes. En 2020, Howard y Gugger describen que las capas se alimentan de la información saliente de las capas que la anteceden, así mismo estas tienen algoritmos internos para que puedan ser entrenadas, como resultado pueden realizar una tarea para la que fue entrenada con mayor exactitud.

El aprendizaje profundo es flexible y simple lo que lo convierte en una técnica con poder, lo que la hace aplicable en diferentes áreas tales como física, medicina, finanzas, arte, entre otras. Un claro ejemplo del impacto que ha tenido el aprendizaje profundo es evidente en la empresa Enlitic dedicada a la medicina, en la misma se emplea este método para diagnosticar enfermedades como consecuencia del análisis de las dolencias. Su creador Jeremy detalla que a los meses de haber sido fundada su empresa con su algoritmo, son más precisos para detectar tumores malignos comparado con los radiólogos.

2.2.9 Medidas de rendimiento

Las evaluaciones de la red implementadas requieren el análisis de ciertos parámetros para verificar su eficiencia, por lo que se tiene:

- a) Verdadero positivo TP': Es aquel modelo que detecta acertadamente el objeto o la categoría para el que fue entrenado
- b) Verdadero negativo TN: Por lo contrario, pero que también demuestra un buen funcionamiento es que el modelo detecta la falta de uno de los objetos a lo que considera la categoría negativa.
- c) Falso positivo FP: el modelo detecta equívocamente el objeto o la categoría para el que fue entrenado.

- d) Falso negativo FN: el modelo o sistema no detecta acertadamente el objeto o la categoría para el que fue entrenado

Estos resultados se los muestra en una matriz llamada matriz de confusión en donde se muestran todos estos parámetros.

2.2.10 Python 3.9.0

“Python es un lenguaje de alto nivel interpretado” (Phyton, 28 de agosto 2022), se enfatiza por la legibilidad en su filosofía de diseño, en este lenguaje, son compatibles la programación orientada a objetos junto con la estructurada. Python utiliza una escritura dinámica, posee una combinación de conteo de referencias, además administra la memoria mediante un recolector de basura que detecta ciclos. Durante la ejecución del programa usa un enlace tardío es decir una resolución dinámica de nombres en la ejecución del programa en el que vincula los métodos y los nombres de las variables.



Figura 15. Página de inicio Python, 2022. Información obtenida de la página oficial de Python. Elaborado por autor

2.2.10.1 OpenCV

Es una biblioteca de enlace de Python diseñada para resolver problemas de visión por ordenador. En comparación con lenguajes como 'C / C ++', 'Python' es más lento. Dicho esto, Python puede ampliarse fácilmente con 'C / C ++', lo que permite escribir código computacionalmente intensivo en C / C ++ y crear envoltorios de Python que puedan ser utilizados como módulos 'Python'. Por lo que se tienen dos ventajas: rapidez igual al del código original porque es el mismo código solo que trabajando en segundo plano, y la otra ventaja es que es un lenguaje muy fácil para programar. OpenCV es un revestimiento de Python para poner en marcha OpenCV C++ original.

2.2.10.2 Pytorch

PyTorch compila información para Python y los programas de este, por lo tanto, fomenta los programas de aprendizaje profundo. Con esta receptividad y conveniencia que se encuentran en este para el aprendizaje profundo y la visión computarizada, las cuales son técnicas expertas para entrenar redes neuronales avanzadas, PyTorch es útil para desarrollar redes neuronales profundas. Tiene un alcance expansivo y se aplica para diversas aplicaciones. Como Python es para la programación, PyTorch es tanto un prólogo magnífico para el aprendizaje profundo como un instrumento utilizable en aplicaciones competentes del mundo real.

2.2.10.3 Numpy

Es una librería numérica de Python utilizada para trabajar con arrays. También tiene funciones para trabajar con álgebra lineal, transformadas de Fourier y matrices.

2.2.11 Anaconda

Anaconda es un reparto libre en la internet tanto de Python como R para la extraer información provechosa de datos que tiene como objetivo ser más eficaz en la gestión y la implementación de paquetes. En Anaconda también es posible gestionar las actualizaciones y versiones de paquetes por el sistema de administración de paquetes, conda, que analiza el entorno actual antes de ejecutar una instalación para evitar la interrupción de otros marcos y paquetes.



Figura 16. Interfaz de inicio del navegador de Anaconda, 2022.
Información obtenida de la aplicación Anaconda, Elaborado por autor

La distribución de Anaconda contiene más de 250 paquetes que vienen incluidos desde que se instala, pero en caso de necesitar otras, es posible instalar muchos paquetes, una cantidad que supera los 7500 de código abierto adicionales desde PyPI, así como el

paquete conda y el administrador de entorno virtual. Además, incorpora una GUI (interfaz gráfica de usuario), Y para tener otra alternativa gráfica para la codificación tiene Anaconda Navigator, Anaconda Navigator está incluido en la distribución de Anaconda, así es posible que los usuarios puedan ejecutar aplicaciones, ejecutar los entornos, paquetes y canales de conda sin usar líneas de comandos. Su navegador (Navigator) puede explorar paquetes, instalarlos dentro de un entorno, ejecutarlos, buscar nuevas actualizaciones y aplicarlas.

2.2.12 Jupyter

Es un software que permite introducir texto formateado, líneas de código, imágenes, vídeos, audios y muchos tipos de multimedia. Jupyter fue premiado con el ACM (Software System Award), gracias a su utilidad, versatilidad y aplicación en muchísimas áreas de interés, es uno de los lenguajes preferidos cuando se desea desarrollar o usar un modelo de aprendizaje profundo.

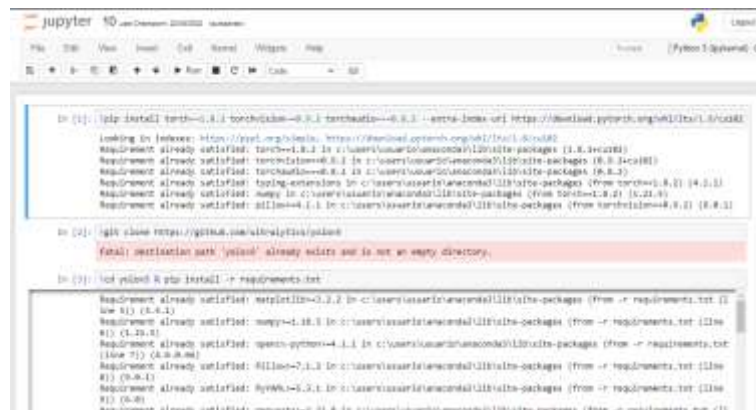


Figura 17. Interfaz de trabajo de Jupyter, 2022. Información obtenida de Jupyter Notebook, Elaborado por autor

2.2.13 Google Colab



Figura 18. Interfaz de bienvenida en Google Colab, 2022. Información obtenida de Google Colab. Elaborado por autor

Google colaboratory, pero siendo popular con su nombre Google Colab, propone el servicio a todo aquel que tenga una cuenta de Gmail, Google le ofrece el uso de código abierto, proporciona una unidad de procesamiento gráfico, siendo una gran opción para

aquellos que no cuentan con los recursos suficientes para el desarrollo de un proyecto, pone a disponibilidad de los usuarios memoria RAM (12,72 GB), con un tiempo de ejecución de 12 horas un disco duro (3548,27 GB).

2.2.14 YOLO

YOLO (you only look once), que sus siglas en español significan solo miras una vez, es un programa que se basa en un modelo de detección de objetos que une en una sola red diferenciable la predicción de cuadros limitantes junto con la clasificación de objetos

La red YOLO posee tres piezas principales.

- 1) Backbone: una RNC que incorpora e indica formas y características de la imagen de entrada en diferentes apariencias.
- 2) Cuello: cadena de capas encaminadas a mezclar y coordinar las características de las imágenes del set de datos, de manera que estos datos sirvan en la predicción.
- 3) Cabeza: consume las características del cuello y realiza los pasos de predicción de cuadro y clase.

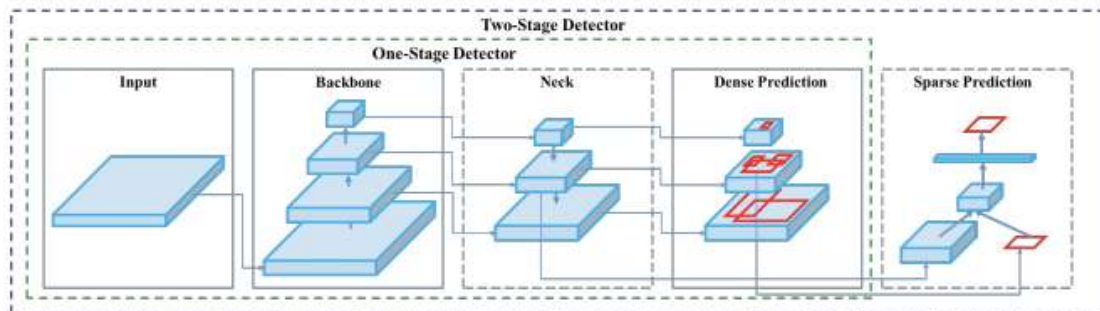


Figura 19. Estructura del funcionamiento de las capas en YOLO. Información obtenida por International Journal of Research in Engineering and Technology. Elaborado por Rakshab Varadharajan, Priyansh Shashikant, Kevin Prabhulal,

2.2.14.1 YOLOv5

YOLOv5 es un lanzamiento reciente de la familia de modelos YOLO. Fue escrito en PyTorch lo que lo hace menos pesado, y mucho más práctico. Con respecto a la arquitectura en red YOLOv5 comparado con su versión anterior YOLOv4, se puede especificar que no hay grandes cambios.

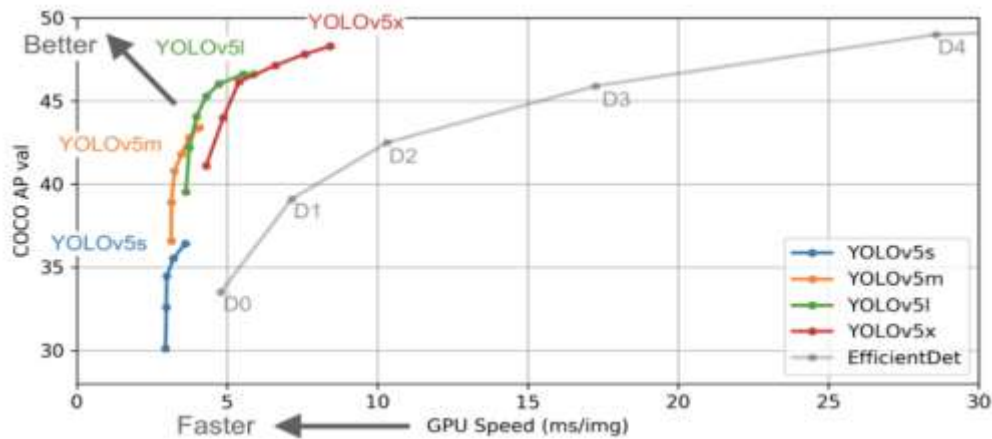


Figura 20. Comparación del funcionamiento de la familia YOLO, 2021. Información obtenida en *International Journal of Research in Engineering and Technology*, Elaborado por Rakkshab Varadharajan, Priyansh Shashikant, Kevin Prabhulal

En el gráfico, el objetivo es producir un modelo de detector de objetos que sea muy eficaz (eje Y) en relación con su tiempo de inferencia (eje X). Los datos preliminares demuestran cuán bien funciona YOLOv5 comparado con otras técnicas igual de modernas.

En resumen, YOLOv5 deriva la mayor parte de su mejora de rendimiento de los procedimientos de entrenamiento de PyTorch.

2.2.14.2 MobileNet

Es una arquitectura que le da la oportunidad al desarrollador de poder cambiar el tamaño de las imágenes, esto como resultado del uso de convoluciones desintegradas, además solo se ingresan dos hiperparámetros generales para así poder reducir su peso, mejorar su latencia y precisión. Por lo que al ser más ligera el desarrollador de un sistema fácilmente puede cambiar la dimensión más adecuada para su modelo.

El modelo de MobileNet contiene una convolución en profundidad, esta es la encargada de filtrar todos los canales de entrada, por lo contrario, una convolución puntual busca realizar una convolución de 1×1 y así mezclar lo que se obtiene como salida de la convolución profunda. En otro punto la convolución estándar destila y coordina las entradas en un grupo inédito para colocar las respuestas en un solo espacio. Ahora una convolución separable segmenta este paso en dos capas, la primera se encarga de depurar y la segunda por lo contrario combina. MobileNetV1 tiene dos capas, que son la capa de convolución en profundidad para el filtrado ligero utilizando solo un filtro convolucional para los canales de entrada y una capa de convolución 1×1 (o puntual) para construir nuevas características mediante el cálculo de combinaciones lineales de los canales de entrada, mientras que

MobileNetV2 consta de dos bloques, que son un bloque residual con una franja de 1 y otro bloque con una franja de 2, que se utilizan para reducir el tamaño. Cada uno de estos bloques tiene 3 capas: una capa de convolución 1×1 con una unidad lineal rectificada (ReLU6), una convolución en profundidad y otra capa de convolución 1×1 sin no linealidad.

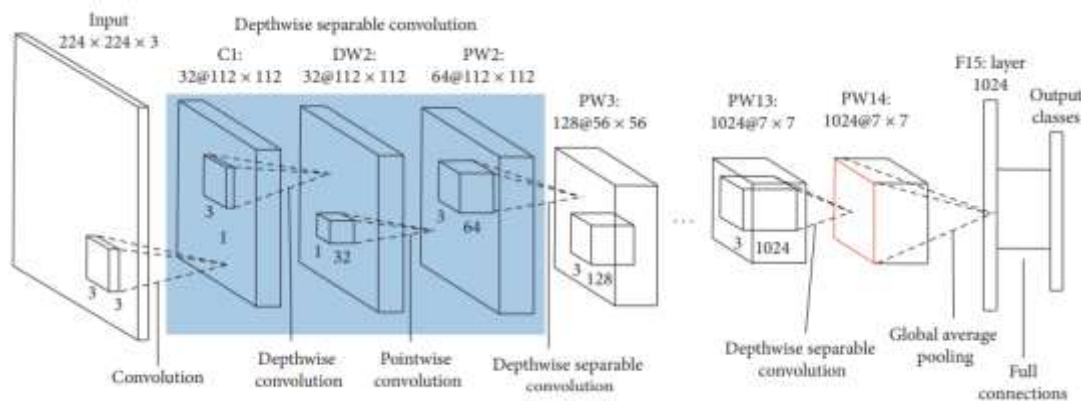


Figura 21. Capas de arquitectura de MobileNet, 2017. Información obtenida del repositorio de Cornell University. Elaborado por Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W. Weyand, T., Andreetto, M., Adam H.

2.2.14.3 Tensorflow

Tensorflow se diseñó pensando en la portabilidad, lo que permite que estos gráficos de cálculo se ejecuten en una diversidad de entornos y plataformas de hardware. La portabilidad de Tensorflow se evidencia al verificar que una misma red neuronal con una codificación semejante se puede entrenar en la nube, después ser distribuida en un clúster de diversas máquinas o a su vez en un solo computador o laptop de interés.

Puede desplegarse para servir predicciones en un servidor dedicado o en plataformas de dispositivos móviles como Android o iOS, o en ordenadores de placa única Raspberry Pi. Tensorflow también es compatible, por supuesto, con los sistemas operativos Linux, macOS y Windows.

Aparte de su portabilidad, otro aspecto clave de Tensorflow es su flexibilidad, que posibilidad a los investigadores y desarrolladores de datos expresar modelos con relativa facilidad. Tensorflow proporciona herramientas útiles para utilizar estos bloques modulares, combinadas con una API flexible que permite escribir otros nuevos. En el aprendizaje profundo, las redes se entrenan con un proceso llamado retro propagación basado en la optimización del descenso de gradiente.

Tensorflow admite de forma flexible muchos algoritmos de optimización, todos con diferenciación automática: el usuario no necesita especificar ningún gradiente de antemano,

ya que Tensorflow los deriva automáticamente basándose en el gráfico de cálculo y la función de pérdida proporcionados por el usuario. Para monitorizar, depurar y visualizar el proceso de entrenamiento, y para agilizar los experimentos, Tensorflow viene con TensorBoard, una sencilla herramienta de visualización que se ejecuta en el navegador.

2.2.14.4 Detector de objetos

Es en donde se buscan los detalles en común de las imágenes que ingresan para su posterior uso en las entradas de las otras capas junto con un sistema de predicción para identificar los objetos etiquetados y aprender sus características y clases.

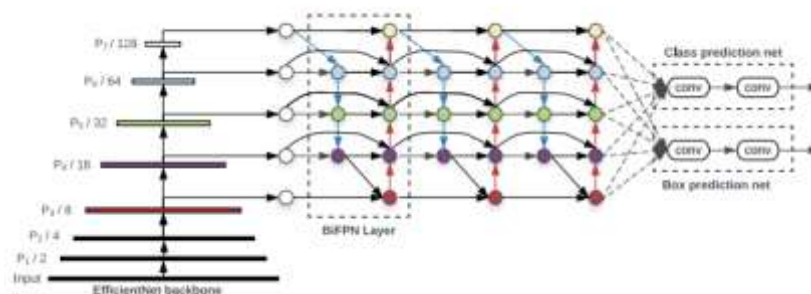


Figura 22. Proceso para detección de objetos, 2019. Información obtenida de Towards Data Science. Elaborado por Tony Shin.

2.2.14.5 LabelImg

Es una herramienta disponible dentro de la internet, al mismo tiempo es gratuita. Tiene su interfaz gráfica diseñada en QT designer, junto con el lenguaje de programación de Python. En general, esta herramienta facilita el etiquetado de centenas de imágenes según las clasificaciones que se quieran realizar. LabelImg permite que los archivos de puedan guardar de tipo VOC, XML, e incluso YOLO (Github, 2022)

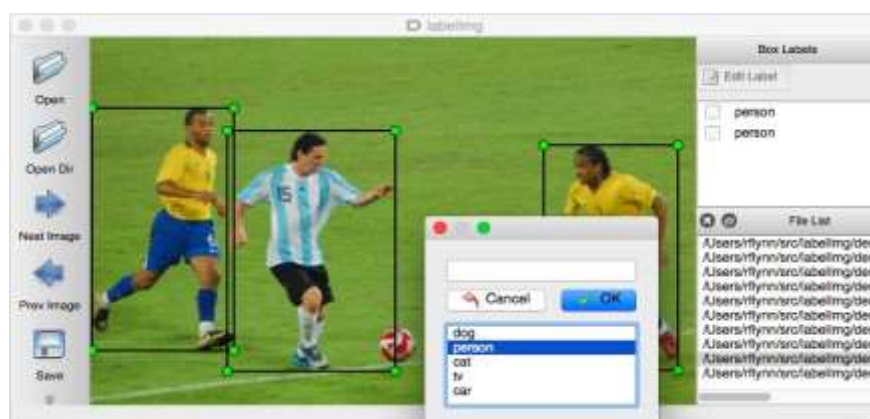


Figura 23. Interfaz de trabajo en LabelImg, 2022. Información obtenida del software LabelImg., Elaborado por autor.

2.3 Marco conceptual

2.3.1 Algoritmo

Es un conjunto de métodos en donde se proporciona el o los procesos adecuados para que un programa resuelva un problema identificado. En general los algoritmos tienen dos puntos primordiales, los cuales son: su diseño, y su eficacia.

- **Diseño:** se evalúa de forma finita varios métodos, pasos o secuencias con los que se puedan dar solución al problema.
- **Eficacia:** es posible que más de un algoritmo resuelva el problema, es por esto que en este punto se considera el tiempo y los recursos que necesita cada uno, y es allí en donde se determina, el algoritmo que lo hace con mayor eficiencia.

2.3.2 Dataset

Un conjunto de datos conocido como dataset pueden ser imágenes, correos electrónicos, indicadores financieros, sonidos, etc. Al ser más común el desarrollo de redes neuronales, con aprendizaje profundo en la internet se encuentran un conjunto de datos disponibles para reducir el tiempo de adquirir estos datos. Muchos de estos conjuntos de datos son creados por académicos para ayudar a avanzar en la investigación.

2.3.3 Anotación de datos

La anotación de datos es la técnica a través de la cual etiquetamos los datos para que las máquinas puedan reconocer los objetos, son el requisito básico para entrenar modelos, la anotación ayuda a reconocer datos relevantes

2.3.4 Etiquetado de datos

El etiquetado de datos se trata de agregar más información/metadatos a varios tipos de datos (texto, audio, imagen y video) para entrenar modelos de aprendizaje automático. El etiquetado tiene que ver con la identificación de características relevantes en el conjunto de datos o cual ayuda a reconocer patrones para entrenar algoritmos.

CAPÍTULO III

DESARROLLO DE LA PROPUESTA

3.1 Metodología de investigación

Con el objetivo de obtener información relevante directamente de los técnicos del laboratorio clínico VIDASSAN, se aplicó el método cualitativo para conocer acerca del control del uso de los equipos de protección personal (EPP). Estos datos son los que sustentan, justifican la investigación y creación del sistema lo que permite validar su realización.

3.2 Metodología del proyecto

En este proyecto se usó la metodología incremental para cumplir con el sistema que detecte los EPP.

3.2.1 Metodología de desarrollo del proyecto

Siendo motivo principal el desarrollo del sistema de detección de EPP en tiempo real se usó la metodología incremental del desarrollo de software, se definieron 4 etapas fundamentales:

En el incremento uno, se definió que la herramienta a emplear para la ejecución del sistema es con Google Colab, por lo descrito en el capítulo dos, es de gran utilidad para el aprendizaje automático, ejecutando la gran cantidad de imágenes con mayor rapidez en comparación con Jupyter Notebook, se puso a prueba el algoritmo en estas herramientas, pero la eficiencia de Google Colab permitió que se obtuvieran resultados más rápidos y efectivos.

En el segundo incremento se tomaron 1025 fotos en las que se observaban todos los equipos de protección personal, que para la presente investigación son los guantes, y la mascarilla.

Como tercer incremento, se llevó a cabo el proceso de entrenar la red neuronal por modelos, como en nuestro caso se van a detectar tres equipos, se entrenaron 3 modelos para cada uno de estos, se subieron los dataset de acuerdo a cada entrenamiento.

Al finalizar cada entramiento, y lo que es el último incremento se realizaron pruebas en tiempo real para verificar que el sistema esté funcionando, y a futuro sea aplicable para el laboratorio VIDASSAN.

3.3 Enfoque de la investigación

3.3.1 Enfoque cualitativo

Se definió el uso de la metodología cualitativa porque se va a definir aquellos equipos de protección personal indispensables para laborar dentro del laboratorio, para así también conocer cómo se está manejando la seguridad ocupacional. Así mismo conocer la opinión de los futuros usuarios sobre la implementación de un sistema con inteligencia artificial, también su punto de vista para conocer cómo les gustaría que se les detectara y se les avisara en el caso de que les falte un EPP.

3.3.2 Operacionalización de variables

En este proyecto se han definido dos variables a partir de la formulación del problema:

- El sistema de detección de EPP en tiempo real se establece como la variable independiente.
- Como variable dependiente se estableció el uso de los EPP dentro del laboratorio.

3.4 Población y muestra

3.4.1 Población

Se focaliza en el noroeste de Guayaquil, en el sector de la Alborada donde se encuentra ubicado una de las sucursales del laboratorio VIDASSAN, en la misma hay 10 colaboradores fuera de los del departamento administrativo, esta información fue proporcionada por el encargado del personal.

3.4.2 Muestra

Al ser pequeña la cantidad de colaboradores dentro del laboratorio VIDASSAN, se realizará la muestra con los 10 técnicos de laboratorio que trabajan en dicha empresa.

3.4.3 Procesamiento y análisis

Con el uso de herramientas informáticas tales como Microsoft Excel y Google Forms se respondieron, y organizaron las respuestas que los colaboradores dieron, y el encargado del personal también, el cuestionario que se usó en Google Forms es de opciones múltiples para las encuestas, y preguntas abiertas para el encargado del personal lo que permite un mejor análisis de los datos al limitar las posibles respuestas, esto beneficiará a que las decisiones que se van a tomar sean más acertadas, para satisfacer las solicitudes que los técnicos realicen.

3.4.4 Técnicas de recolección de datos

Las encuestas serán la técnica a emplear, siendo el cuestionario su instrumento. Como se mencionó previamente se usó Google Forms, esta herramienta también permite recibir las respuestas de los usuarios, por lo tanto, es por este medio en el cual se obtuvieron las respuestas, reduciendo el uso del papel y se evitó un acercamiento más o contacto directo lo que también permitió reducir el riesgo de contagio ya que nuestro país aún atraviesa por una pandemia.

3.4.5 Técnicas estadísticas para el procesamiento de la información

Cuando Google Forms mostraba las preguntas y recibía las respuestas, automáticamente las analizaba y representaba en un diagrama de pastel graficando el porcentaje para cada respuesta, incluso indica la frecuencia absoluta pero también muestra las relativas.

3.4.6 Análisis de resultados

Como fruto de las respuestas de las encuestas efectuadas, se ha determinado que los trabajadores son conscientes del alto riesgo y peligro al que se exponen a diario el 90% asegura entender cuán peligroso es el ambiente en donde laboran a diario y aún más sin el uso de los equipos de seguridad personal. Son conscientes de que el laboratorio cumple con sus obligaciones para salvaguardar su seguridad, la mayoría aseguró que los propietarios del laboratorio si les proporcionan los EPP.

Un punto considerable para el desarrollo de este proyecto que se denotó tras el análisis de las contestaciones de los colaboradores es que ellos indicaron con un 100%

como se muestra en el anexo 48 que dentro de los equipos de protección que más usan durante su labor en el laboratorio se encuentran los guantes y la mascarilla con un porcentaje bastante alto está la bata como lo puede visualizar en el mismo anexo. Estas respuestas sustentan que los equipos de protección personal que se ha escogido para identificar con el sistema de inteligencia artificial son los más usados, por lo tanto, este será de utilidad para VIDASSAN.

Se les consultó sobre la frecuencia de uso de la bata, e indicaron que a veces la usan depende de la actividad que realicen porque cabe recalcar que en el laboratorio también realizan análisis de rayos x, y tomografías en donde no es obligatorio el uso de la bata.

Con referencia al seguimiento que se le da al cuidado y uso de los EPP, una cantidad mayor a la media evidencia que el supervisor del personal realiza una revisión constante entre cada 2 o más horas, por lo que ellos también estiman de buena manera la implementación de un sistema con inteligencia artificial para realizar la identificación de estos. Sugieren también que este sistema muestre en recuadros la detección de estos equipos que le facilitarán estar seguros de que fueron detectados, y si no tienen alguno de estos, la mayoría aspira que se les presente un mensaje en la pantalla, todo esto se evidencian en los anexos 50 hasta 59.

Producto de la entrevista realizada al señor Ángel Vera Acosta, encargado del personal del laboratorio VIDASSAN, comprende la importancia de los EPP para que los técnicos laboren y así precautelar peligros en su salud y seguridad, dentro de sus actividades principales está en verificar que ellos estén usando los EPP indispensables según el área en el que laboren cumpliendo un reglamento interno que tienen, en general, indicó que desconoce sobre un sistema que detecte este tipo de equipamiento, y por último considera que los propietarios si estarían dispuestos a implementar a largo plazo el sistema que se propone.

3.5 Propuesta tecnológica

En este proyecto investigativo se busca cumplir la meta principal que es el desarrollo de un sistema con inteligencia artificial para el procesamiento de vídeo en la detección del uso de equipos de seguridad personal de los técnicos del laboratorio clínico VIDASSAN.

El desarrollo de este sistema planteado intenta concientizar a los colaboradores del laboratorio clínico del uso de los EPP necesarios para laborar dentro del mismo, una vez que se haya detectado la falta de algún equipo de bioseguridad en la pantalla de la computadora portátil se mostrará aquel que hace falta, pero también se mostrará todo EPP detectado. La herramienta Google Colab, junto con el sistema Tensorflow son los encargados de simular a través de vídeos el uso y la falta de los EPP, mientras la laptop procesa el vídeo para mostrarlo en pantalla, los programas y todas las técnicas que se programaron se les va aplicando para que detecte a través de visión artificial.

3.5.1 Análisis de factibilidad

El sistema de detección de EPP para el laboratorio VIDASSAN en donde se realiza la detección a través del procesamiento de vídeo es la solución que se presenta para la supervisión de este tipo de equipos de protección personal y así ayudar a reducir riesgos, al controlar que todos los operarios dentro del laboratorio estén bien equipados.

3.5.2 Factibilidad técnica

El sistema planteado es accesible, al ser una propuesta, usa tecnología y técnicas que no requieren grandes recursos técnicos, el uso de software libre representa también un beneficio en la ejecución de este sistema. Con respecto al hardware se emplea un laptop de índole personal pero que puede ser reemplazada a futuro por una adquirida por el laboratorio. Con lo antes mencionado, se detallarán a continuación las condiciones básicas de hardware y software para el desarrollo del sistema.

Tabla 3. Hardware usado para el desarrollo

Equipo	Especificaciones
Laptop Dell	Laptop Dell Inspiron 15 5567
	Windows 10 Home
	Memoria RAM: 8 GB
	Procesador: Intel® Core™ i7-7500U CPU @ 2.70GHz 2.90 GHz
	Tarjeta gráfica: Intel ® HD Graphics 620
	Resolución de cámara integrada 640x480 pixeles

Información tomada de la página oficial de Dell, Elaborado por el autor.

Tabla 4. *Software usado para el desarrollo*

Componente	Descripción
Sistema Operativo	Windows 10 / 64 bits
	Python
	Versión: 3.6.13
Lenguaje de programación	Tensorflow
	Versión: 2.6
IDE	Visual Studio Code
	Versión: 1.70.2
Herramienta de anotación de imágenes	LabelImg
	Versión: 1.8.6
Plataforma Web	Google Colab

Información tomada del sistema creado. Elaborado por el autor.

3.5.3 Factibilidad legal

En base a las leyes vigentes en el Ecuador, el sistema no incumple ninguna de estas, en virtud de usar un software libre por lo que no se viola algún derecho de autor o propiedad intelectual, además se enfoca en la seguridad operacional que se le tiene que brindar al técnico de laboratorio para el cumplimiento de sus labores.

Se infiere entonces que el sistema si es factible legalmente. La fundamentación legal se detalla en el anexo 3, en este apartado se indican los artículos u ordenanzas que sustentan este proyecto investigativo que como objetivo tiene precautelar la seguridad física y biológica del trabajador

3.5.4 Factibilidad económica

Como se lo había mencionado previamente para el desarrollado del sistema se usó software libre, además el hardware es propiedad del autor, sin embargo, se generaron ciertos gastos.

A continuación, se indican los costos para el desarrollo del sistema.

Tabla 5. *Detalle de los costos para la elaboración del sistema*

Hardware	
Laptop Dell	\$1025
Recursos humanos	
Desarrollador	\$1500 / 3 meses
Otros gastos	
Electricidad	\$50
Suministros de oficina	\$10
Transporte	\$40
Alimentación	\$210
Total	\$2,835

Información obtenida del desarrollo del sistema, Elaborado por el autor.

Por lo mostrado y descrito anteriormente, se puede concluir que este sistema es rentable para el laboratorio, además cuando lo deseen implementar dentro de las instalaciones del laboratorio, no requerirán de una mayor inversión.

3.6 Desarrollo de la propuesta

3.6.1 Requerimientos funcionales

En esta sección se describen los requerimientos funcionales, entre estos se mencionan los detalles técnicos de cómo se procesarán los datos y otras funcionalidades que se cumplen con el sistema.

Tabla 6. Descripción de un requerimiento funcional del sistema

ID	RF_01
Nombre	Detección en tiempo real de EPP
Tipo	Exigencia operativa
Prioridad	Alta
Prerrequisito	N/A
Descripción	El sistema a partir de que una persona aparezca en la cámara empieza a detectar los EPP para los que fue entrenada
Entrada	Dataset de las fotos etiquetadas
Proceso	Detección de guantes, mascarilla de laboratorio
Salida	Identificación de que el técnico tenga todos los EPP básicos que se solicitan para laborar y salvaguardar su bioseguridad.

Información obtenida del desarrollo del sistema, Elaborado por el autor.

Tabla 7. Segundo requerimiento funcional

ID	RF_02
Nombre	Rastreo de los EPP
Tipo	Exigencia operativa
Prioridad	Alta
Prerrequisito	RF_01
Descripción	Una vez que ya haya detectado los EPP en el técnico, aunque este realice movimientos ligeros, tales como mover la cabeza o las manos el sistema debe ser capaz de identificar a los objetos en la nueva posición.
Entrada	Imagen en tiempo real del técnico
Proceso	Seguimiento de guantes y mascarilla dentro del laboratorio
Salida	Detección de los EPP en la nueva ubicación

Información obtenida del desarrollo del sistema, Elaborado por el autor.

Tabla 8. Tercer requerimiento funcional del sistema

ID	RF_03
Nombre	Indicar que faltan EPP
Tipo	Exigencia operativa
Prioridad	Alta
Prerrequisito	RF_01
Descripción	Cuando el sistema esté ejecutando y la persona esté frente a la cámara, se empiece con el proceso de detección y no encuentre alguno de los objetos para los cuales este fue entrenado por lo que de forma visual va a indicar que está incompleto.
Entrada	Técnico sin el uso de uno o varios de los EPP
Proceso	No haber identificado ninguno de los equipos básicos de protección personal para los que fue entrenado.
Salida	Indicación de que faltan los EPP

Información obtenida del desarrollo del sistema, Elaborado por el autor.

3.6.2 Requerimientos no funcionales

3.6.2.1 Comprensible

El sistema a crear en donde está disponible la visualización de la cámara es de fácil acceso y uso, en donde se empleará una interfaz sencilla tanto para el encargado de verificar que los técnicos estén bien equipados como para los usuarios finales.

3.6.2.2 Operatividad

Cuando se desee mostrar los EPP que fueron detectados como aquellas partes en donde no se detectaron los objetos, deberá mostrar en la pantalla los recuadros para que sea visible el hecho de que si se les detecto de forma adecuada la protección que están usando.

Así también se tienen las siguientes exigencias no operacionales

Tabla 9. *Exigencias no operacionales del sistema*

Exigencias no operacionales	Título	Descripción
<i>ENO_01</i>	Respuesta de tiempo tolerable	El sistema al ejecutarse no debe retardar al mostrar la detección del objeto.
<i>ENO_02</i>	Sistema práctico para los usuarios	Para la ejecución del sistema se busca que sea práctico para el manejo del supervisor a cargo de la verificación del uso de los EPP

Información obtenida del desarrollo del sistema, Elaborado por el autor.

3.6.3 Requisitos para operatividad del software

Para cumplir con todas las exigencias y suplir las necesidades por las cuales fue diseñado este sistema es necesario lo siguiente:

Tabla 10. *Requisitos de operatividad del software para el sistema*

Requisito	Función
<i>Os</i>	Permiten manipular la estructura de directorios (para leer y escribir archivos)
<i>Uuid</i>	Es un método seguro que permite generar identificadores únicos fuertes cifrados a través de nodos
<i>Path</i>	Son configuraciones de los directorios donde estarán almacenadas nuestras carpetas
<i>Wget</i>	Ayuda a descargar archivos directamente desde el servidor principal.
<i>Tensorflow</i>	Gran herramienta que ayuda a crear, implementar, y entrenar modelos para detectar objetos para los que se entrena.
<i>CV2</i>	Permite identificar objetos, caras, clasificar acciones.

Información obtenida del desarrollo del sistema, Elaborado por el autor.

3.6.4 Sujetos del sistema

Dentro del sistema hay varios sujetos que participen de la estructura del sistema, además de los actores que intervienen, en la tabla 11 se detallarán cada uno de ellos junto con su descripción.

Tabla 11. *Sujetos del sistema de detección de EPP en tiempo real*

Sujeto	Detalle
<i>Técnico de laboratorio</i>	Es el sujeto de interés el mismo que usará el sistema que se desarrolló.
<i>Sistema</i>	Programa que se ocupa de la detección en tiempo real de la mascarilla y los guantes a través de la cámara integrada de una laptop.
<i>Técnica</i>	Es la herramienta más importante y prioritaria con el que se usará la mejor técnica de redimensionar las imágenes para agilizar la detección.
<i>Detección</i>	Es el que se ocupa de identificar o detectar tanto la mascarilla como los guantes.
<i>Aviso</i>	Junto con la programación del sistema y la falta de detección de los EPP, es en donde se aplicará el aviso visual para notificar a los técnicos.

Información obtenida del desarrollo del sistema, Elaborado por el autor.

3.6.5 Diagrama de casos de usos

3.6.5.1 Diagrama de caso de uso del reconocimiento de los guantes y mascarilla

Como se puede observar en la figura 24, se detalla el caso de uso cuando un técnico del laboratorio aparezca en el recuadro de la cámara para empezar con la detección, y en cuanto el sistema detecte los EPP el proceso que dará, indicando el proceso operativo que realiza.

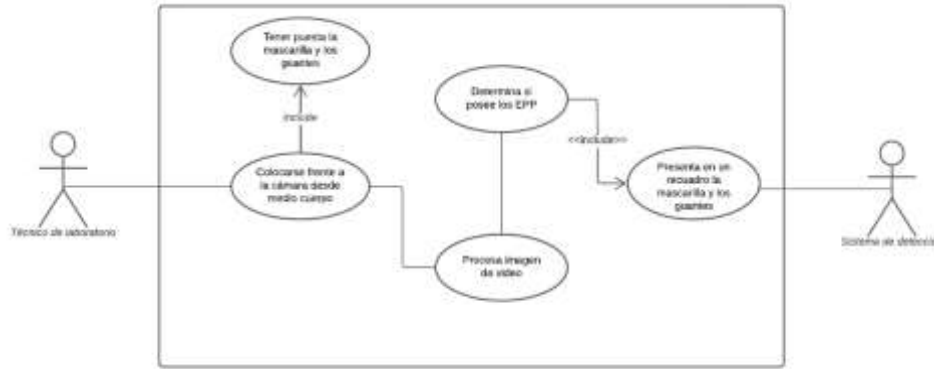


Figura 24. Caso de uso identificación de mascarilla y guantes, 2022, *Información obtenida del autor, Elaborado por el autor.*

3.6.5.2 Diagrama de caso de uso del seguimiento de la mascarilla

A continuación, se muestra el segundo caso de uso en donde una vez detectada la mascarilla, se va a realizar el seguimiento del objeto a través de la cámara integrada en la laptop, en donde se debe seguir mostrando la mascarilla detectada.



Figura 25. Segundo caso de uso para el seguimiento de mascarilla. 2022, *Información obtenida del autor, Elaborado por el autor.*

3.6.5.3 Diagrama de caso de uso del seguimiento de los guantes

El tercer caso es parecido al anterior, pero sin duda representa un reto mayor para detectarlos ya que el sistema deberá buscar en la misma imagen los dos guantes, y al ser las manos mucho más despegables, tendrá que buscarlo en todo el recuadro.



Figura 26. Tercer caso de uso para el seguimiento de guantes. 2022, *Información obtenida del autor, Elaborado por el autor.*

3.6.5.4 Diagrama de caso de uso cuando no se detecta los EPP

Proceso que realiza el sistema cuando este no detecta alguno de los EPP, indica la falta de mascarilla y guante

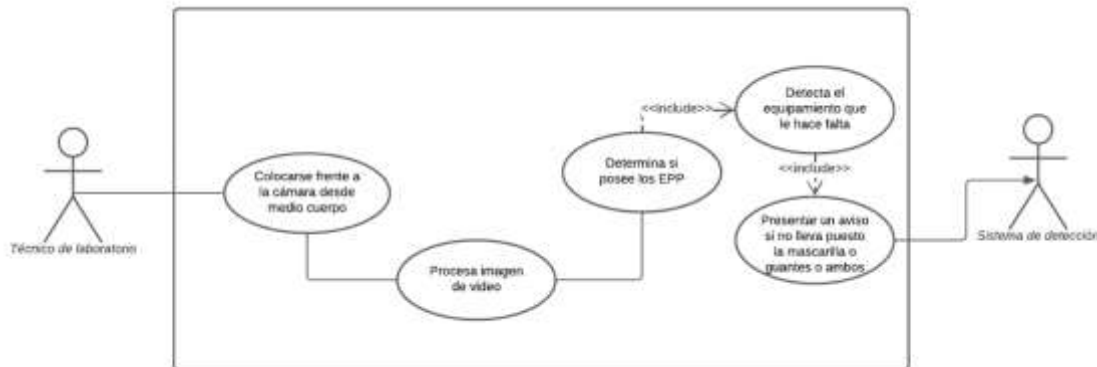


Figura 27. Diagrama de caso de uso al no detectar los EPP. 2022, Información obtenida del autor, Elaborado por el autor.

3.6.6 Desarrollo del sistema

3.6.6.1 Proceso de selección de herramienta

Anteriormente se mencionó que este proceso fue parte del primer incremento para el desarrollo de este sistema. Es importante describir el proceso que se siguió para determinar cuál es la herramienta más indicada para elaborar y crear el sistema.

3.6.6.1.1 Problemas presentados al usar Jupyter Notebook

Durante un tiempo, mayor a un mes y medio se intentó realizar el sistema usando la herramienta Jupyter Notebook, se entrenó la red neuronal con 1025 imágenes, cada una de estas etiquetadas con ambos equipos de seguridad personal que son la mascarilla y los guantes.

Una vez determinado el dataset se ejecutó el código, inicialmente con 500 épocas, las imágenes fueron redimensionadas a 320x320, y se usó la librería YOLO5. Con esta cantidad de fotos, el sistema cargaba cada época en 23 minutos, es decir que en una hora a penas se ejecutaban un poco más de 2 épocas, por consiguiente, al día se obtenían 48 épocas entrenadas, en general, para realizar un solo entrenamiento se necesitaban 24,984 horas lo que representan 10 días y medio.

Una de las razones por las que se presentaba este problema inicialmente es por el equipo que se usó para crear el sistema, la falta de capacidad en las tarjetas gráficas y de espacio en RAM, generaba gran lentitud para entrenar la red neuronal.

El tiempo invertido en el entrenamiento sería considerado fructífero, pero los resultados no fueron los esperados. Una vez que finalizó el entrenamiento, se probaron ingresando imágenes que servían como testeo para el sistema, leía la imagen, pero este no reconocía los objetos, cuando se abría la cámara para detectar en tiempo real, etiquetaba otro objeto ajeno a los de interés.

Al no encontrar resultados favorables y óptimos con un entrenamiento de 500 épocas, se intentó entrenar el sistema con 400 épocas más, es decir que en ese punto la red neuronal fue entrenada con 900 épocas, este proceso fue con el objetivo de verificar si era directamente proporcional la cantidad de épocas con cuán precisa es en la detección.

En este último intento con la herramienta Jupyter Notebook, usando la memoria RAM de la laptop y su GPU, se ejecutó el entrenamiento, el cual necesitó 46 minutos entre cada época, por lo que en este entrenamiento a penas se podía obtener una época por hora, y para la finalización de todas las épocas fueron necesarios 37 días y medio, la carga fue muy lenta e incluso así tampoco hubieron buenos resultados, porque cuando se ejecutaba el código para activar la cámara, y una persona aparecía no reconocía los objetos, seguía presentando inconvenientes para detectarlos.

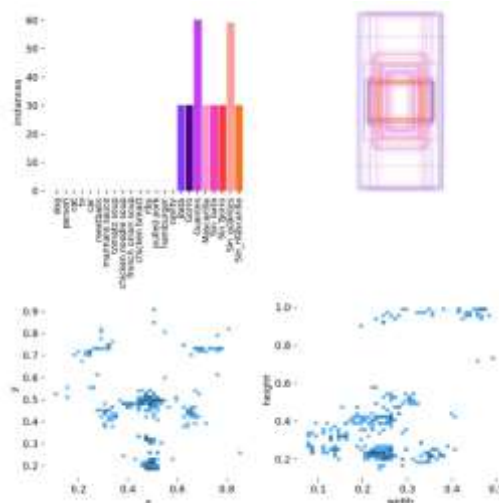


Figura 28. Resultados de train con Jupyter Notebook y YOLO, 2022, Información obtenida del autor, Elaborado por el autor.

Con lo expuesto y teniendo en consideración la limitante de tiempo para el desarrollo de este proyecto, se decidió buscar una nueva alternativa, es por esto que para el desarrollo de este sistema se usó la herramienta Google Colab, que trabaja de la misma manera que Jupyter Notebook, pero que agrega otras ventajas y beneficios tales como que al emplear la opción que propone del entrenamiento en la GPU este es 60 veces más rápido que en la CPU, consecuentemente el entrenamiento, es mucho más rápido en comparación si se usara Jupyter Notebook directamente instalada en una computadora o laptop.

3.6.6.2 Creación del dataset de la recolección de imágenes

Para entrenar la red neuronal dentro del algoritmo creado en Google Colab es necesario que se tenga con anticipación un conjunto de imágenes en donde se observe cada uno de los objetos a detectar, a este conjunto de imágenes se lo conoce como el dataset. Es sustancial recopilar imágenes en donde se pudieran apreciar los guantes, y la mascarilla.

Por políticas de seguridad empleadas dentro del laboratorio clínico VIDASSAN no se pudieron tomar fotos de los técnicos dentro del laboratorio, por este motivo se tomaron fotos que sirvieron para el entrenamiento del sistema.

Es importante destacar que se estableció como regla, tener una cantidad mayor a 100 imágenes o fotos para el entrenamiento del sistema, particularmente en este proyecto se usaron 1025 fotos, para asegurar que el entrenamiento genere mayor asertividad para detectar los objetos.

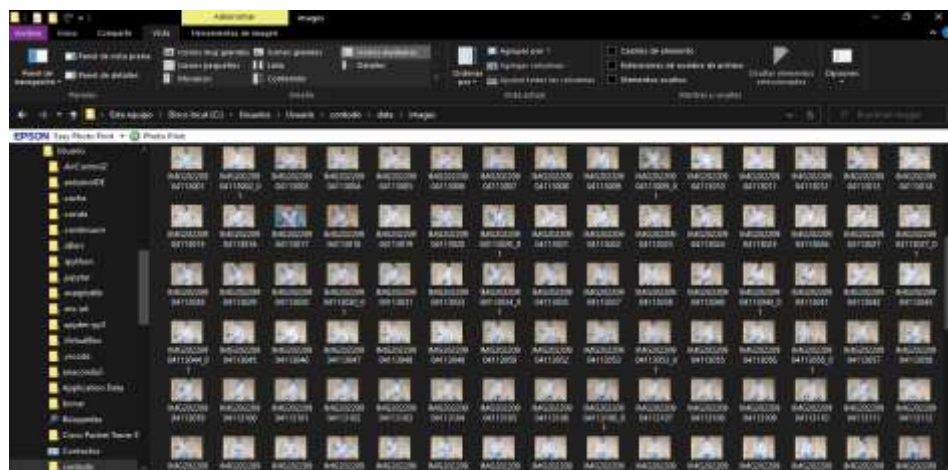


Figura 29. Evidencia de las imágenes del dataset. 2022, Información obtenida del autor, Elaborado por el autor.

Necesitamos las imágenes en sí mismas, y sus etiquetas, estas proceden de un conjunto limitado de categorías (mascarilla y guante). Además, se consideró que la cantidad de imágenes para cada categoría debe ser pareja, las mismas etiquetas para mascarilla como para guantes. Por lo que respecta a lo antes indicado se cumple, ya que en las mismas imágenes están etiquetadas los dos equipos de protección. A modo de ejemplo en este proyecto no se usaron más imágenes de guantes que de mascarillas, porque si se tiene el doble de imágenes de un equipo que, del otro, entonces el detector estará naturalmente sesgado o sobre ajustado en las categorías fuertemente representadas. El desequilibrio de clases es un problema común en el aprendizaje automático.

3.6.6.3 Estandarización de las imágenes

Se planificaron varios preparativos necesarios para la correcta formación del modelo una vez preparado el dataset para acelerar el proceso.

- El sistema convierte todo el dataset de imágenes a escala de grises, posteriormente previo a la presentación de los resultados con la librería cv2, realizará nuevamente la transformación de grises a la escala RGB.
- Se determinó cambiar el tamaño de las imágenes a 320x320 pixeles
- Para que la detección genere resultados más acertados y también teniendo en consideración que el sistema será empleado en un laboratorio clínico se estipuló que las fotografías o imágenes que sirvieron de dataset tuvieran un fondo claro y monocromático con el fin de evitar confusión mientras se entrenaba la red neuronal.
- Anotar en el conjunto de datos las etiquetas, encuadrando los objetos de interés.



Figura 30. Carpetas de datos en Google Colab, 2022, Información obtenida del autor, Elaborado por el autor.

3.6.6.4 División del dataset

Para el entrenamiento de un sistema con detección de objetos es recomendable dividir el dataset, porque hay de dejar un grupo de imágenes que servirán para las pruebas del sistema, estas imágenes el modelo que se crea nunca las ha visto, de allí es por donde se empieza a evaluar si este funciona, y si detecta los objetos para los que fue entrenado, aprendiendo a partir del segundo grupo que son los de la sección de train o entrenamiento. En lo que se refiere a ambos grupos, es trascendental que no dependa un grupo del otro, ni del paquete de entrenamiento del de pruebas, ni viceversa.

Para el sistema de detección de EPP (mascarillas y guantes) en tiempo real se usará el 80% de las imágenes para el entrenamiento o train, y el 20% para las pruebas o test, esta elección fue de forma aleatoria puesto que en cada imagen están los dos objetos detectados a la vez.

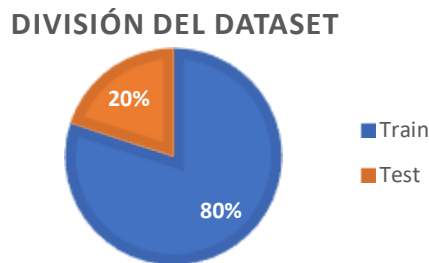


Figura 31. Gráfico de pastel de la división del dataset. 2022, Información obtenida del autor, Elaborado por el autor.

Tabla 12. Distribución de imágenes del dataset

Dataset	Número de fotos
Imágenes para entrenamiento	820 fotos
Imágenes para pruebas	205 fotos
Total	1025 fotos

Información obtenida del autor, Elaborado por el autor.

3.6.6.5 Labels

Los labels que son conocidos como las etiquetas que usamos para señalar en nuestra imagen y que así la red neuronal aprenda se lo realizó en la herramienta labelImg.

Desde el código básico para su uso en Jupyter Notebook en donde se clona su librería y todos los módulos internos que la integran para su funcionamiento.

En cada imagen que forma parte del dataset se etiquetaron la mascarilla y los guantes, pero estos fueron etiquetados de forma singular, a cada mano se le etiquetó el guante, no se lo realizó de forma plural ya que podría generar confusión durante el entrenamiento. LabelImg reduce el tiempo de repetir el proceso de crear directorios solo con un tipo de etiqueta, por lo contrario, como se va a mostrar en las imágenes permite que se etiqueten más de un tipo de objeto a la vez.

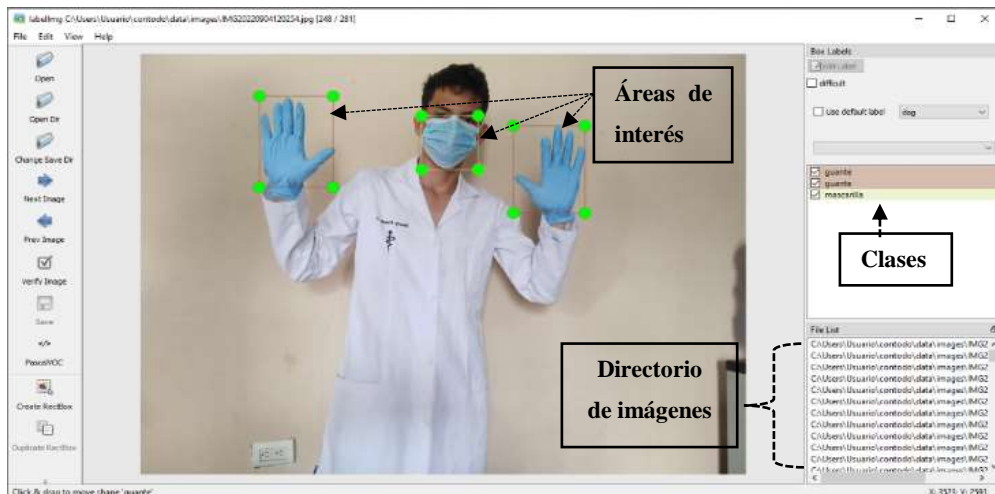


Figura 32. Etiquetas de una foto del dataset. 2022, Información obtenida del autor, Elaborado por el autor.

3.6.6.6 Selección de la arquitectura para el sistema

Como se advirtió previamente el proceso de entrenamiento de un aprendizaje profundo requiere de una gran cantidad de horas, al ser este proyecto investigativo desarrollado en un tiempo máximo de 3 meses, es necesario buscar alternativas con el fin de mejorar la asertividad de respuesta del sistema a desarrollar, es por esto que después de las pruebas ejecutadas de forma directa con Jupyter Notebook lo que trajo como consecuencia la reducción del tiempo general, se buscó un modelo pre-entrenado, estos tipos de modelos ya poseen una serie de capas de entrenamiento que son funcionales y editables según sea necesario, puesto que estos vienen entrenados para reconocer objetos de forma general y que para nuestro caso no es aplicable, más sin embargo, su programación y entrenamiento previo lo son.

Se eligió la RNC (Red Neuronal Convolutacional) MobileNetV2 como clasificador. Se eligió esa RNC tomando en consideración los beneficios que la identifican como son

el poco tiempo de inferencia, y su muy buena precisión para detectar los objetos. MobileNetV2 representa una versión mejorada a comparación de su versión anterior. la arquitectura de la red MobileNetV2 tiene 19 bloques o conocidos como los residuos del cuello de botella.

3.6.6.7 Proceso de entrenamiento de RNC

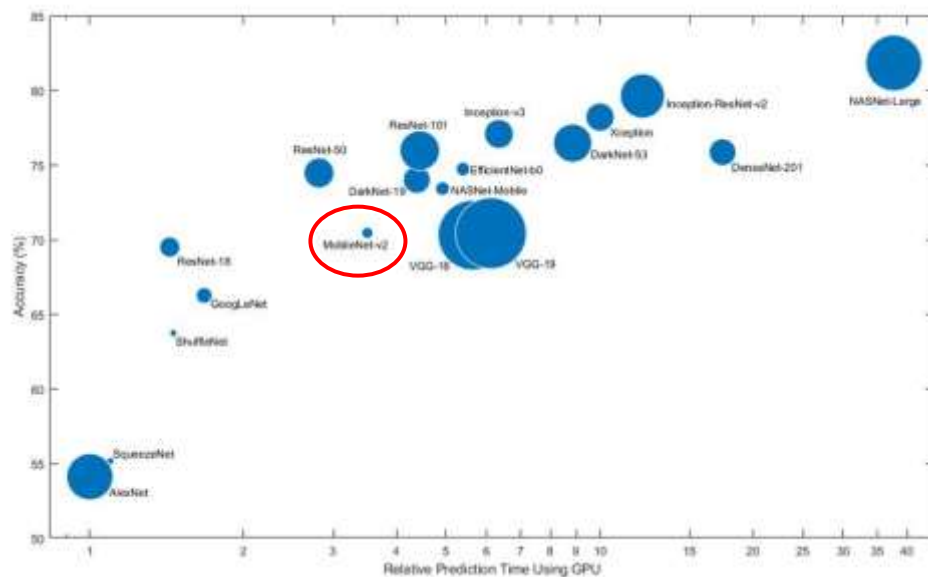


Figura 33. Comparación de los modelos pre-entrenados, 2022, Información obtenida en la página oficial de MathWorks, Elaborado por MathWorks

Se muestra la eficiencia media del modelo MobileNetV2 en comparación con otros, su porcentaje de asertividad es de aproximadamente el 70%, y no requiere tanto tiempo usando el GPU. La forma de entrenar un modelo de detección de objetos comienza con la adquisición de datos. Aunque existen muchas formas de automatizar el proceso, en este caso específico de capacitación, la adquisición de datos se realizó manualmente y de forma personalizada, al tomar las fotos y filtrar las imágenes ruidosas y garantizar la calidad del conjunto de datos de capacitación.



Figura 34. Imagen que genera ruido para el sistema, Información obtenida del autor, Elaborado por el autor.

Una vez que se adquieren los datos, las imágenes en el conjunto de datos deben etiquetarse. El etiquetado de imágenes se completó con el software LabelImg, que es una implementación de código abierto basada en Python.

El resultado generado de los esfuerzos de etiquetado de imágenes es un archivo XML que abarca los cuadros delimitadores de las imágenes etiquetadas. En la figura 34 se muestra el formato de este tipo de archivos. Sin embargo, los modelos de redes neuronales no pueden analizar archivos XML y requieren un formato específico llamado Tensor Records, por lo que es necesario convertirlos a este formato, este proceso se lleva a cabo dentro del código anexo.

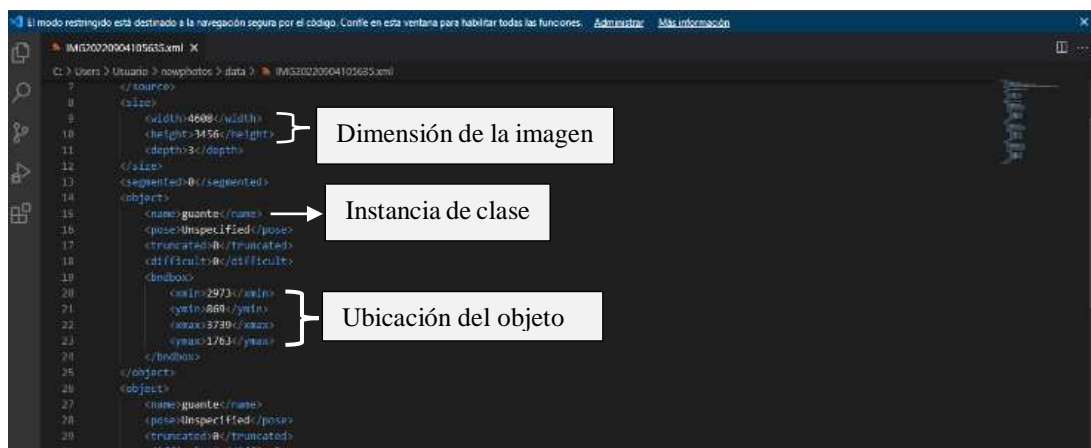


Figura 35. Captura de pantalla de la estructura de un archivo xml, 2022, Información obtenida del autor, Elaborado por el autor.

Al crear el archivo TF-Records, se debe generar el mapa de etiquetas. El mapa de etiquetas es esencialmente un codificador para las clases que se entrenarán, ya que la red neuronal predice solo en números después de la capa sigmoidea final. Para nuestro caso son dos clases: mascarilla y guante.

La configuración del entrenamiento implica la selección del modelo, así como los parámetros del modelo, el tamaño del lote, la cantidad de pasos, el entrenamiento y el directorio de registros TF de prueba. Afortunadamente, la canalización de detección de objetos de Tensorflow viene con un archivo preconfigurado que optimiza la mayor parte de la selección de configuración para un entrenamiento eficiente del modelo.

Hay varios parámetros que deben actualizarse. En primer lugar, se debe configurar la ruta al mapa de etiquetas o el archivo de texto de detección de objetos que

se creó anteriormente para que el modelo sepa qué etiqueta colocar en la imagen una vez detectada.

A continuación, se debe proporcionar la ruta a los archivos de entrenamiento y prueba de TF-Record. El número de clases lo va a establecer el programa una vez que lea los archivos e identifique a través del código en el anexo 9, para este caso se etiquetaron dos objetos por lo que el mismo sistema lo definió en dos y se estableció un tamaño de lote de 4 para la capacitación. La cantidad de pasos, que se refiere simplemente a la cantidad de épocas de entrenamiento, se establece en 25000. Luego, se descarga el modelo SSD MobileNet V2 y la ubicación del archivo de punto de control también se incorpora en el archivo de configuración.

Una vez realizado los procesos de entrenamiento para la detección de la mascarilla y los guantes y la falta de los mismos, cada clase con un modelo por separado, pero que se fusionan posteriormente para la detección, es cuando se empieza con el desarrollo de los avisos en cuanto no se detecte alguno de los equipos, se va a notificar con un mensaje en la pantalla y un ícono para alertar a los colaboradores de que les falta algún EPP o todos, aparecerá un mensaje que dice “Equipamiento incompleto”, en la sección de pruebas se muestra esta interfaz.

Es de gran importancia la notificación para que los colaboradores tengan precaución y no se expongan frente peligros físicos como biológicos

3.6.7 Etapa de pruebas

Como todo desarrollo, no se puede obtener resultados óptimos, sin tener modelos que presenten bajos resultados, estos nos sirven también para evaluar las características de los modelos, e ir variando los parámetros para recabar mejores resultados.

En definitiva, realizar el proceso de pruebas permite alcanzar una de las metas definidas del proyecto

Después de entrenar el modelo, se debe verificar su rendimiento. Esto se hace al probar el rendimiento del modelo en datos nunca antes vistos. Los datos no vistos utilizados son el conjunto de prueba en el que se dividieron los datos anteriormente. Si la prueba se realizara con los mismos datos que se usaron para el entrenamiento, no se obtendrá una medida precisa, ya que el modelo ya está acostumbrado a los datos y

encuentra los mismos patrones que antes. Esto le dará una precisión desproporcionadamente alta.

Cuando se utiliza el conjunto de datos test para realizar las pruebas, se obtiene una medida más precisa del rendimiento del modelo y su velocidad.

Como se muestra en la figura 35 usando 469 fotos, batch size de 4 imágenes y solo 3000 train steps, se obtuvieron bajos porcentajes de asertividad como se puede apreciar.

```
Anaconda Prompt (Anaconda3) - python \Users\rober\Documents\Trabajo\Tesis\Tesis_2\Tesis\Tensorflow\models\research\object_det
Evaluate annotation type "bbox"
DONE (t=2.68s).
Accumulating evaluation results...
DONE (t=0.20s).
Average Precision (AP) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.693
Average Precision (AP) @[ IoU=0.50 | area= all | maxDets=100 ] = 0.975
Average Precision (AP) @[ IoU=0.75 | area= all | maxDets=100 ] = 0.893
Average Precision (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.693
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 1 ] = 0.587
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 10 ] = 0.749
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.751
Average Recall (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
Average Recall (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
Average Recall (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.751
INFO:tensorflow:Eval metrics at step 3000
10910 16:45:04.067680 33208 model_lib v2.py:1015] Eval metrics at step 3000
```

Figura 36. Captura de pantalla de una de las pruebas realizadas al sistema. Información obtenida del autor, Elaborado por el autor.

Con estos parámetros el porcentaje más alto que se obtuvo fue de 75%, y solo detectaba la mascarilla, es decir que el sistema era obsoleto para la detección de los guantes.

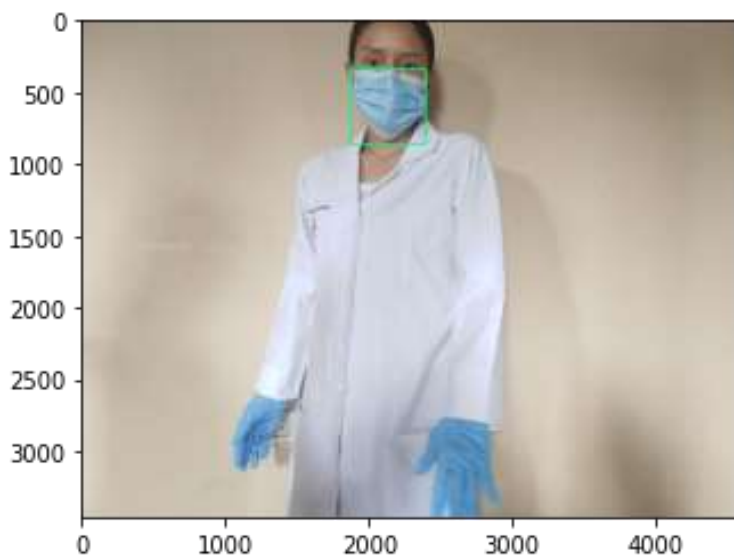


Figura 37. Sistema sin detectar guantes, 2022, Información obtenida del autor, Elaborado por el autor.

En otra prueba realizada se pudo obtener mejores resultados, más sin embargo la detección de los guantes seguía sin ser efectiva, en este nuevo caso se usaron 756 fotos, 5000 train steps, con el mismo batch size.

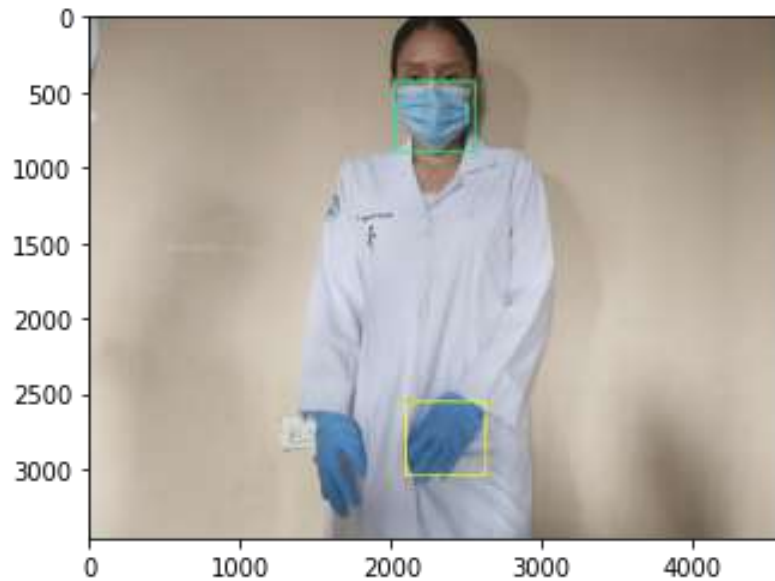


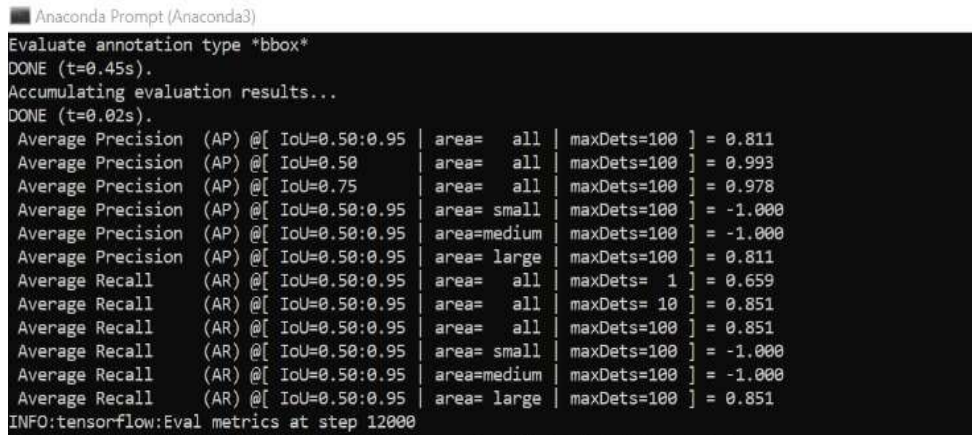
Figura 38. Sistema que tiene poca asertividad para detectar guantes
Información obtenida del autor, Elaborado por el autor.

Un modelo que presentó más éxito en la detección de EPP fue el que tenía los siguientes parámetros, 25000 train steps, un batch size de 4, y un conjunto de imágenes del dataset de 820 fotos.

```
Instructions for updating:
Use fn_output_signature instead
[0910 17:17:40.691638 140473080803392 deprecation.py:560] from /usr/local/lib/python3.7/dist-packages/tensorflow/python
Instructions for updating:
Use fn_output_signature instead
INFO:tensorflow:Step 2100 per-step time 1.414s
[0910 17:20:11.049535 140478917347200 model_lib_v2.py:707] Step 2100 per-step time 1.414s
INFO:tensorflow: {'loss/classification_loss': 0.092960574,
'loss/localization_loss': 0.063488616,
'loss/regularization_loss': 0.14308485,
'loss/total_loss': 0.29952604,
'learning_rate': 0.07090056}
[0910 17:20:11.058121 140478917347200 model_lib_v2.py:708] {'loss/classification_loss': 0.092960574,
'loss/localization_loss': 0.063488616,
'loss/regularization_loss': 0.14308485,
'loss/total_loss': 0.29952604,
'learning_rate': 0.07090056}
INFO:tensorflow:Step 2200 per-step time 0.868s
[0910 17:21:37.668060 140478917347200 model_lib_v2.py:707] Step 2200 per-step time 0.868s
INFO:tensorflow: {'loss/classification_loss': 0.18136327,
'loss/localization_loss': 0.08835359,
'loss/regularization_loss': 0.14243114,
'loss/total_loss': 0.412148,
'learning_rate': 0.07088167}
[0910 17:21:37.668918 140478917347200 model_lib_v2.py:708] {'loss/classification_loss': 0.18136327,
'loss/localization_loss': 0.08835359,
'loss/regularization_loss': 0.14243114,
'loss/total_loss': 0.412148,
'learning_rate': 0.07088167}
```

Figura 39. Captura de pantalla del proceso de entrenamiento, 2022, Información obtenida del programa Google Colab, Elaborado por el autor.

Una vez identificado el mejor modelo, donde se obtienen mejores resultados se procedió a crear las condiciones con el objetivo de que se comunique por medio de un mensaje en la pantalla, el código de las condiciones se encuentra en los anexos.



```

Anaconda Prompt (Anaconda3)
Evaluate annotation type *bbox*
DONE (t=0.45s).
Accumulating evaluation results...
DONE (t=0.02s).
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.811
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.993
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.978
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.811
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.659
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.851
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.851
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.851
INFO:tensorflow:Eval metrics at step 12000

```

Figura 40. Resultados de modelo con mejores resultados, 2022, Información obtenida de Google Colab, Elaborado por el autor.

3.6.7.1 Condición cuando se identifican todos los EPP

Haciendo hincapié que solo se van a detectar los guantes y la mascarilla que son parte del conjunto de los EPP que usan los técnicos de laboratorio del laboratorio VIDASSAN.

Cuando el sistema detecte que la persona está usando ambos equipos de seguridad, en la pantalla aparecerá un mensaje con un ícono, indicando que el equipamiento está completo como se muestra en la siguiente figura la lógica implementada para este caso y el posterior.

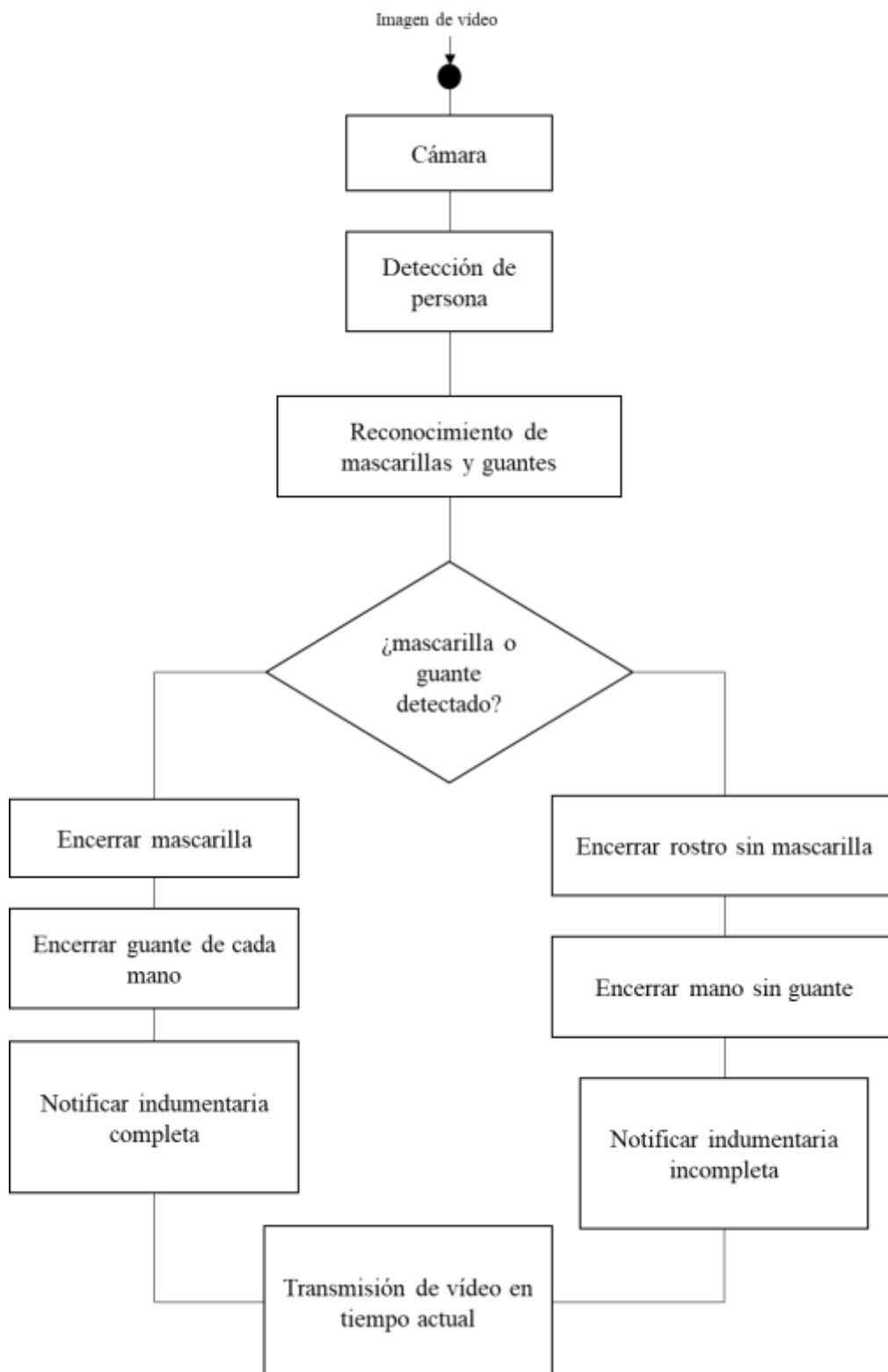


Figura 41. Diagrama de bloques del sistema propuesto, 2022, Elaborado por el autor.

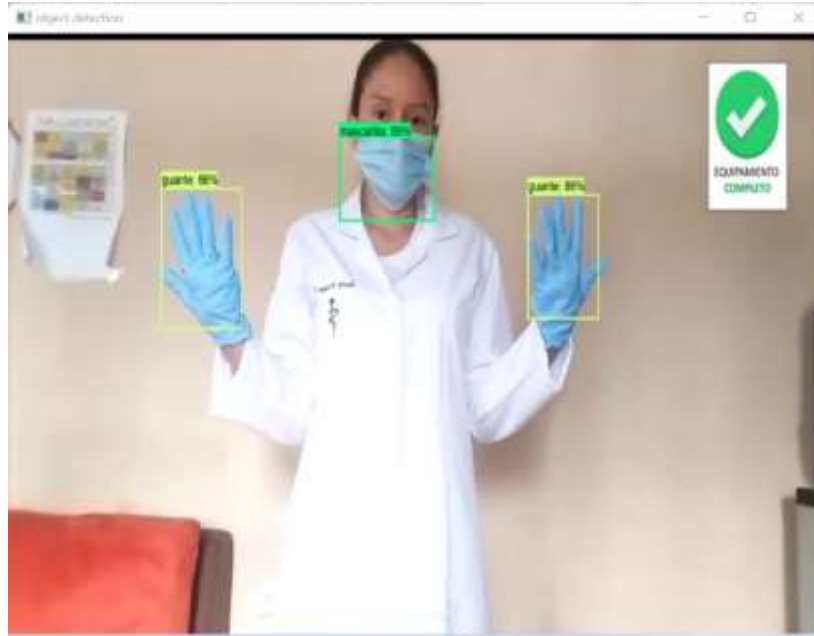


Figura 42. Interfaz de equipamiento completo, 2022, Información obtenida del sistema desarrollado, Elaborado por el autor.

3.6.7.2 Condición cuando no se detecten todos los EPP

Para indicarle a los colaboradores dentro del laboratorio que no está completo su equipamiento también se condicionó al sistema para que les notificara, no solo cuando no detecte los guantes ni la mascarilla, sino también cuando detecte la mascarilla y no los guantes o viceversa.



Figura 43. Interfaz de equipamiento incompleto, 2022, Información obtenida del sistema desarrollado, Elaborado por el autor.



Figura 44. Detectar “sin_mascarilla”, 2022, Información obtenida del sistema desarrollado, Elaborado por el autor.

3.6.7.1 Resultados de matriz de confusión

En tensorboard herramienta de Tensorflow para generar la matriz de confusión no toma en consideración los falsos negativos, solo toma los positivos verdaderos (TP), los falsos positivos (FP), y los falsos negativos (FN). En el modelo de la mascarilla se pueden observar los resultados en la tabla 13, y el modelo de los guantes se observan en la tabla 14. Consecutivamente, con estos valores se determinará la precisión (P) del sistema tanto para la detección de la mascarilla, sin mascarilla, guantes y sin guantes, en la ecuación se toman los verdaderos positivos se dividen para la suma de estos valores con los falsos positivos:

$$P = \frac{Tp}{Tp + Fp}$$

Tabla 13. Matriz de confusión del modelo de mascarillas

		Valor Verdadero	
		Clase Positiva	Clase negativa
Predicción	Clase Positiva	32 (TP)	10 (FP)
	Clase negativa	9 (FN)	-

Información obtenida del sistema desarrollado, Elaborado por el autor.

Tabla 14. *Matriz de confusión del modelo de sin mascarillas*

		Valor Verdadero	
		Clase Positiva	Clase negativa
Predicción	Clase Positiva	33 (TP)	9 (FP)
	Clase negativa	10 (FN)	-

Información obtenida del sistema desarrollado, Elaborado por el autor.

Tabla 15. *Matriz de confusión del modelo de guantes*

		Valor Verdadero	
		Clase Positiva	Clase negativa
Predicción	Clase Positiva	41 (TP)	4 (FP)
	Clase negativa	6 (FN)	-

Información obtenida del sistema desarrollado, Elaborado por el autor.

Tabla 16. *Matriz de confusión del modelo de sin guantes*

		Valor Verdadero	
		Clase Positiva	Clase negativa
Predicción	Clase Positiva	35 (TP)	4 (FP)
	Clase negativa	9 (FN)	-

Información obtenida del sistema desarrollado, Elaborado por el autor.

En el modelo para la predicción de mascarillas se obtiene una precisión de 0.91, para el modelo de sin mascarillas 0.78, para el modelo de guantes 0.89, y para el modelo de sin guantes con una precisión menor en comparación a otros se obtuvo 0.76.

3.7 Conclusiones

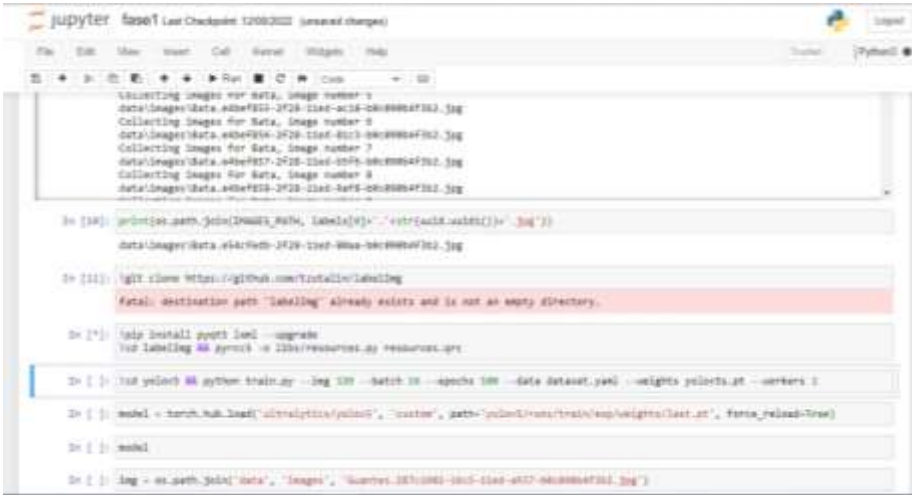
- Se desarrolló un sistema capaz de detectar en tiempo real dos de los implementos más usados por los técnicos de laboratorio, tales como los guantes y mascarilla, con el uso de una red neuronal convolucional y de usar el modelo pre-entrenado SSD MobileNet v2, logrando así un porcentaje de asertividad poco más del 90% para la mascarilla y un 89% de precisión en la detección de guantes.
- Google Colab representó una gran herramienta para el desarrollo no solo de este sistema sino cualquier entrenamiento de un sistema de detección de objetos, especialmente durante la etapa de entrenamiento, al poner en disposición una unidad de procesamiento gráfico de gran tamaño, y al ser capaz de expandir la memoria RAM para procesar un amplio grupo de información, sea texto, imágenes o vídeos.
- Al buscar detectar objetos en un ambiente específico como las instalaciones del laboratorio VIDASSAN, fue de gran ayuda la decisión de tomar fotos de autoría con un fondo monocromático, facilitó que el entrenamiento de la red neuronal convolucional sea mucho más efectiva y precisa al mostrar los resultados.
- Filtrar las imágenes que no mostraban los objetos de forma clara para el entrenamiento también aportó a mejorar los resultados, así como el emplear la herramienta labelImg y etiquetar de forma manual los guantes y mascarilla.
- Emplear la arquitectura SSD MobileNet v2 posibilita que no se requiere de grandes recursos para su entrenamiento y funcionamiento, lo que trae como resultado, que este sistema sea de fácil uso en otra computadora sin que esta requiera gran disponibilidad de procesadores o memoria.
- El uso del conjunto de imágenes detectadas que forman el dataset del sistema con el modelo pre-entrenado, el cual detecta algunos objetos por defecto, permitió también que se detectaran los EPP con un alto porcentaje de asertividad.
- Cada parámetro con el que se entrena un modelo genera grandes cambios, desde la cantidad de train steps o pasos de entrenamientos que se deseen realizar, así como el tamaño del lote con el que se va a sustentar el sistema, y ciertamente la cantidad de imágenes con las que el sistema va a aprender representa también un punto crítico para la ejecución de cada modelo.
- Se demuestra cuán útil es el sistema cuando este notifica en la pantalla el equipamiento incompleto

3.8 Recomendaciones

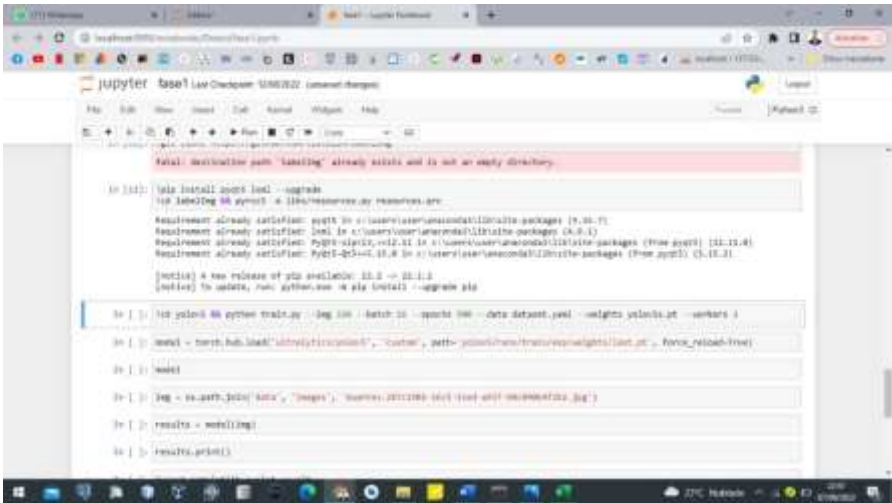
- Comprender la parte teórica de cómo se estructuran las redes neuronales y sus tipos, permite que se pueda escoger apropiadamente cual de estos es más conveniente usar según el tipo de detención de objetos que se aspira realizar.
- Es eficiente descargar las últimas versiones de los programas y herramientas que se van a emplear para el entrenamiento de un sistema de detección, tomando como ejemplo al modelo MobileNet, este tiene una versión inferior que no ofrece la misma eficiencia y rapidez como lo hace su versión posterior, así mismo Tensorflow, que actualiza de forma recurrente sus comandos, por lo cual es siempre importante verificar sus actualizaciones en las páginas oficiales.
- Es muy proactivo estipular un tiempo para realizar diversas pruebas del sistema, creando y ejecutando diversos modelos, para así al final escoger aquel con mejores resultados.
- Se debe tomar en cuenta tener disponible un gran grupo de imágenes para desarrollar un sistema de detección de objetos para que el sistema pueda aprender mejor los objetos que se desean detectar.
- Hay muchos tipos de objetos que se desean detectar por lo que hay una amplia gama de modelos pre-entrenados, es primordial evaluar cada uno de ellos, y posterior al análisis tomando en consideración cuanto requiere de almacenamiento y la precisión que ofrece el modelo, es cuando se debe llegar a una decisión.
- Para trabajos futuros y que no se tenga una computadora o laptop con un gran procesador, así como reducir tiempo una gran opción es entrenar el detector de objetos con Google Colab
- Por la limitación del tiempo y lo requerido por el laboratorio se estableció solo detectar dos tipos de EPP, pero tomando como base este algoritmo se puede entrenar el sistema para que sea capaz de detectar la bata de laboratorio, gorro desechable, zapatones, o visores.
- La detección de los objetos se la realiza con la cámara integrada de una laptop, para trabajos futuros este sistema se lo puede diseñar para que detecte los objetos con cámara de seguridad internas del laboratorio.

ANEXOS

ANEXOS



Anexo 1. Evidencia de código en Jupyter Notebook. Información obtenida del sistema de prueba, Elaborado por el autor



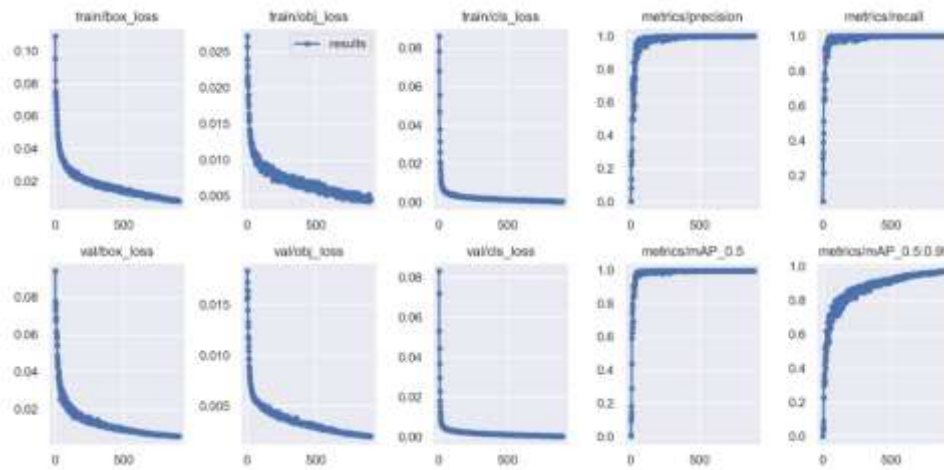
Anexo 2. Código para ingresar imagen de prueba en Jupyter Notebook. Información obtenida del sistema de prueba, Elaborado por el autor

The screenshot displays a Microsoft Excel spreadsheet with a table containing 10 columns and 20 rows of data. The columns are labeled as follows:

- transform_001
- transform_002
- transform_003
- transform_004
- transform_005
- transform_006
- transform_007
- transform_008
- transform_009
- transform_010

The data is organized into a structured format, likely representing a dataset for analysis. The first row of data (row 2) contains values for each of the 10 columns. Subsequent rows (rows 3 through 21) contain numerical data for each column. The values are generally small, ranging from approximately -0.000000 to 0.000000, with some values being slightly negative or positive. The data appears to be organized into a structured format, likely representing a dataset for analysis.

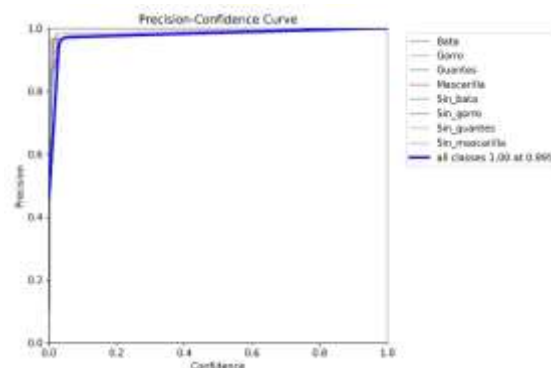
Anexo 3. Resultados de épocas de entrenamiento. Información obtenida del sistema de prueba, Elaborado por el autor



Anexo 4. Resultados gráficos del entrenamiento en Jupyter Notebook. Información obtenida del sistema de prueba, Elaborado por el autor



Anexo 5. Dataset del entrenamiento en Jupyter Notebook. Información obtenida del sistema de prueba, Elaborado por el autor



Anexo 6. Curva de precisión del sistema en Jupyter Notebook. Información obtenida del sistema de prueba, Elaborado por el autor

Desde el primer anexo hasta este, se muestran las evidencias del trabajo de prueba realizado en Jupyter Notebook, es justamente en el anexo 6 donde se muestra que con una curva de 1 no se identifican puntos como positivos, es decir que no hay verdaderos positivos ni falsos positivos, siendo así un sistema que no detecta con precisión los objetos.

Código del sistema

```
In [ ]: !pip install --upgrade pyqt5 lxml

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pyqt5
  Downloading PyQt5-5.15.7-cp37-abi3-manylinux1_x86_64.whl (8.4 MB)
    |████████████████████████████████████████| 8.4 MB 26.3 MB/s
Requirement already satisfied: lxml in /usr/local/lib/python3.7/dist-packages (4.9.1)
Collecting PyQt5-sip<13,>=12.11
  Downloading PyQt5_sip-12.11.0-cp37-cp37m-manylinux1_x86_64.whl (344 kB)
    |████████████████████████████████████████| 344 kB 68.5 MB/s
Collecting PyQt5-Qt5>=5.15.0
  Downloading PyQt5_Qt5-5.15.2-py3-none-manylinux2014_x86_64.whl (59.9 MB)
    |████████████████████████████████████████| 59.9 MB 1.4 MB/s
Installing collected packages: PyQt5-sip, PyQt5-Qt5, pyqt5
Successfully installed PyQt5-Qt5-5.15.2 PyQt5-sip-12.11.0 pyqt5-5.15.7
```

Anexo 7. Instalar PyQt5. Información obtenida del sistema desarrollado, Elaborado por el autor.

Se instala y actualiza PyQt5 ya que es un conjunto completo de enlaces de Python.

```
In [ ]: from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

Anexo 8. Acceder a Drive. Información obtenida del sistema desarrollado, Elaborado por el autor.

Copia de acceso directo al drive

```
In [ ]: import uuid
import os
import time
```

Anexo 8. Importar librerías. Información obtenida del sistema desarrollado, Elaborado por el autor.

Se importa

- 1.- uuid: Un identificador único
- 2.- os: Accede a funcionalidades dependientes del Sistema
- 3.- time: tiempo de respuesta.

```
In [ ]: labels = ["mascarilla", "guante"]
```

Anexo 10. Etiquetas de clases. Información obtenida del sistema desarrollado, Elaborado por el autor.

Etiquetas personalizadas

- 1.- Mascarilla
- 2.- Guantes

```
In [ ]: ROOT_PATH = "/content/drive/MyDrive/Tesis/Tensorflow/"
```

Anexo 11. Ruta de la carpeta. Información obtenida del sistema desarrollado, Elaborado por el autor.

Se pone la ruta de la carpeta, importante que este en posix

Ya que provee una manera versátil de usar funcionalidades dependientes del S.O

```
In [ ]: os.name
```

```
Out[6]: 'posix'
```

Anexo 12. Instalar las funcionalidades dependientes del S.O., Información obtenida del sistema desarrollado, Elaborado por el autor.

```
In [ ]: TRAIN_PATH = os.path.join(ROOT_PATH, 'workspace', 'images', 'train')
TEST_PATH = os.path.join(ROOT_PATH, 'workspace', 'images', 'test')
ARCHIVE_PATH = os.path.join(ROOT_PATH, 'workspace', 'images', 'archive.tar.gz')
```

Anexo 13. Rutas de configuración. Información obtenida del sistema desarrollado, Elaborado por el autor.

Rutas de configuraciones de las carpetas con los archivos

os.path.join: Es una ruta relativa al directorio actual en la unidad

```
In [ ]: !tar -czf {ARCHIVE_PATH} {TRAIN_PATH} {TEST_PATH}
```

```
tar: Removing leading `/' from member names
```

Anexo 14. Leer y escribir archivos tar. Información obtenida del sistema desarrollado, Elaborado por el autor.

Permite leer y escribir archivos tar con el nombre de la variable asignada en las rutas

```
In [ ]: CUSTOM_MODEL_NAME = 'my_ssd_mobnet'
PRETRAINED_MODEL_NAME = 'ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8'
PRETRAINED_MODEL_URL = 'http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8.tar.gz'
TF_RECORD_SCRIPT_NAME = 'generate_tfrecord.py'
LABEL_MAP_NAME = 'label_map.pbtxt'
```

Anexo 15. Modelos dependientes en los directorios. Información obtenida del sistema desarrollado, Elaborado por el autor.

Modelos dependientes de los archivos en los directorios

```
In [ ]: paths = {
    'WORKSPACE_PATH': os.path.join(ROOT_PATH, 'workspace'),
    'SCRIPTS_PATH': os.path.join(ROOT_PATH, 'scripts'),
    'API_MODEL_PATH': os.path.join(ROOT_PATH, 'models'),
    'ANNOTATION_PATH': os.path.join(ROOT_PATH, 'workspace', 'annotations'),
    'IMAGE_PATH': os.path.join(ROOT_PATH, 'workspace', 'images'),
    'MODEL_PATH': os.path.join(ROOT_PATH, 'workspace', 'models'),
    'PRETRAINED_MODEL_PATH': os.path.join(ROOT_PATH, 'workspace', 'pre-trained-models'),
    'CHECKPOINT_PATH': os.path.join(ROOT_PATH, 'workspace', 'models', CUSTOM_MODEL_NAME),
    'OUTPUT_PATH': os.path.join(ROOT_PATH, 'workspace', 'models', CUSTOM_MODEL_NAME, 'export'),
    'TFJS_PATH': os.path.join(ROOT_PATH, 'workspace', 'models', CUSTOM_MODEL_NAME, 'tfjs-export'),
    'TFLITE_PATH': os.path.join(ROOT_PATH, 'workspace', 'models', CUSTOM_MODEL_NAME, 'tflite-export'),
    'PROTOS_PATH': os.path.join(ROOT_PATH, 'protos')
}
```

Anexo 16. Manipulación de rutas locales. Información obtenida del sistema desarrollado, Elaborado por el autor.

Manipulación de rutas locales configuradas como:

- 1.- Rutas del espacio: Es el espacio de trabajo principal
- 2.- Ruta del Scripts: Generación de registro tf de tensorflow
- 3.- Ruta del Modelo API: Descargar la biblioteca del modelo

- 4.- Ruta de Anotación: Muestra el mapa de etiquetas
- 5.- Ruta de imágenes: Aquí es donde estará el etiquetado e imágenes
- 6.- Ruta modelo: Donde irá nuestro entrenamiento y puntos de control para luego restaurar y construir nuestro modelo de detección en tiempo real
- 7.- Ruta del modelo Pre-entrenado: Aquí se aprovechan los modelos de detección de objetos para que previamente se puedan transferir el aprendizaje y básicamente pueda construir nuestro modelo mucho más rápido
- 8.- Ruta del punto de control: Aquí se alojarán los archivos
- 9.- Ruta de salida: Es dentro del modelo en su espacio de trabajo
- 10.- Rutas de protocolos

```
In [ ]: files = {
    'PIPELINE_CONFIG': os.path.join(ROOT_PATH, 'workspace', 'models', CUSTOM_MODEL_NAME, 'pipeline.config'),
    'TF_RECORD_SCRIPT': os.path.join(paths['SCRIPTS_PATH'], TF_RECORD_SCRIPT_NAME),
    'LABELMAP': os.path.join(paths['ANNOTATION_PATH'], LABEL_MAP_NAME)
}
```

Anexo 17. Ruta de los archivos con rutas paths, Información obtenida del sistema desarrollado, Elaborado por el autor.

Rutas de los archivos con enlazamientos de las rutas paths

Os.name == 'posix': método donde se puede encontrar el nombre del S.O, para manipular archivos y directorios

```
In [ ]: for path in paths.values():
    if not os.path.exists(path):
        if os.name == 'posix':
            mkdir -p {path}
        if os.name == 'nt':
            mkdir {path}
```

Anexo 18. Manipular archivos y directorios, Información obtenida del sistema desarrollado, Elaborado por el autor.

os.name == 'nt': Ejecuta Windows y con el 'posix'

```
In [ ]: if os.name=='nt':
    !pip install wget
    import wget
```

Anexo 19. Descargar la herramienta wget, Información obtenida del sistema desarrollado, Elaborado por el autor.

Ejecución del método dentro del sistema

- 1.- Se descarga la herramienta wget para descarga contenidos desde servidor web
- 2.- Se importa


```
In [ ]: if not os.path.exists(os.path.join(paths['APIMODEL_PATH'], 'research', 'object_detection')):
        !git clone https://github.com/tensorflow/models $paths['APIMODEL_PATH']
```

Anexo 20. Clonar el modelo Tensorflow, Información obtenida del sistema desarrollado, Elaborado por el autor.

Se clona el repositorio de tensorflow para almacenarlo en la carpeta

Se accede a la carpeta donde están todos los archivos para el sistema

```
In [ ]: cd "/content/drive/MyDrive/Tesis/"
        /content/drive/.shortcut-targets-by-id/1n8EXG-mf-UwI-W6RixOYD7IXSuFEy9nP7/Tesis
```

Anexo 21. Se accede a las carpetas, Información obtenida del sistema desarrollado, Elaborado por el autor.

```
In [ ]: if os.name=='posix':
        !apt-get install protobuf-compiler
        !cd tensorflow/models/research && protoc object_detection/protos/*.proto --python_out=. && python -m pip install .

if os.name=='nt':
    url="https://github.com/protocolbuffers/protobuf/releases/download/v3.15.6/protoc-3.15.6-win64.zip"
    wget.download(url)
    !move protoc-3.15.6-win64.zip {paths['PROTOD_PATH']}
    !cd {paths['PROTOD_PATH']} && tar -xzf protoc-3.15.6-win64.zip
    os.environ['PATH'] += os.pathsep + os.path.abspath(os.path.join(paths['PROTOD_PATH'], 'bin'))
    !cd tensorflow/models/research && protoc object_detection/protos/*.proto --python_out=. && copy object_detection\packages\it
    !cd tensorflow/models/research/slim && pip install -e .

Reading package lists... Done
Building dependency tree
Reading state information... Done
protobuf-compiler is already the newest version (3.0.0-9ubuntu1).
The following package was automatically installed and is no longer required:
  libnvidia-common-460
Use 'apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 10 not upgraded.
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Processing /content/drive/.shortcut-targets-by-id/1n8EXG-mf-UwI-W6RixOYD7IXSuFEy9nP7/Tesis/Tensorflow/models/research
DEPRECATION: A future pip version will change local packages to be built in-place without first copying to a temporary directory. We recommend you use --use-feature=in-tree-build to test your packages with this new behavior before it becomes the default.
```

Anexo 22. Condicionar el poxis, Información obtenida del sistema desarrollado, Elaborado por el autor.

Se instala el Tensorflow Object Detection

- 1.- Se coloca una condición al poxis que viene hacer para usar funcionalidades dependientes del S.O
- 2.- Se instala el protobuf
- 3.- Se accede a la carpeta y se instala

```
1.1)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.9.0->tf-models-official==2.5.1->object-detection==0.1) (1.14.1)
Requirement already satisfied: tensorflow-estimator<2.10.0,>=2.9.0rc0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.9.0->tf-models-official==2.5.1->object-detection==0.1) (2.9.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.9.0->tf-models-official==2.5.1->object-detection==0.1) (1.1.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.9.0->tf-models-official==2.5.1->object-detection==0.1) (1.0.3)
Requirement already satisfied: gast<0.4.0,>=0.2.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.9.0->tf-models-official==2.5.1->object-detection==0.1) (0.4.0)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.9.0->tf-models-official==2.5.1->object-detection==0.1) (3.3.0)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.9.0->tf-models-official==2.5.1->object-detection==0.1) (3.1.0)
Requirement already satisfied: keras-preprocessing>=1.1.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.9.0->tf-models-official==2.5.1->object-detection==0.1) (1.1.2)
Requirement already satisfied: flatbuffers<2,>=1.12 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.9.0->tf-models-official==2.5.1->object-detection==0.1) (1.12)
```

Anexo 23. Lista de requerimientos en la clonación de Tensorflow, Información obtenida del sistema desarrollado, Elaborado por el autor.

```
In [ ]: if os.name == 'posix':
        wget (PRETRAINED_MODEL_URL)
        mv (PRETRAINED_MODEL_NAME+'.tar.gz') {paths['PRETRAINED_MODEL_PATH']}
        cd {paths['PRETRAINED_MODEL_PATH']} && tar -zxvf (PRETRAINED_MODEL_NAME+'.tar.gz')
    if os.name == 'nt':
        wget.download(PRETRAINED_MODEL_URL)
        move (PRETRAINED_MODEL_NAME+'.tar.gz') {paths['PRETRAINED_MODEL_PATH']}
        cd {paths['PRETRAINED_MODEL_PATH']} && tar -zxvf (PRETRAINED_MODEL_NAME+'.tar.gz')

--2022-09-06 17:26:16-- http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8.tar.gz
Resolving download.tensorflow.org (download.tensorflow.org)... 108.177.13.128, 2607:f8b0:400c:c00::80
Connecting to download.tensorflow.org (download.tensorflow.org)|108.177.13.128|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 28515344 (28M) [application/x-tar]
Saving to: 'ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8.tar.gz'

ssd_mobilenet_v2_fp 100%[=====] 19.5M 79.0MB/s in 0.2s

2022-09-06 17:26:16 (79.6 MB/s) - 'ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8.tar.gz' saved [28515344/28515344]

ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8/
ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8/checkpoint/
ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8/checkpoint/ckpt-0.data-00000-of-00001
ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8/checkpoint/checkpoint
ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8/checkpoint/ckpt-0.index
ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8/pipeline.config
ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8/saved_model/
ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8/saved_model/saved_model.pb
ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8/saved_model/variables/
```

Anexo 24. Acceder al sistema con el posix, Información obtenida del sistema desarrollado, Elaborado por el autor.

Se accede al sistema con el posix y la ruta del directorio para descargar el archivo del modelo pre entrenado.

```
In [ ]: !pip install tensorflow-object-detection-api

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: tensorflow-object-detection-api in /usr/local/lib/python3.7/dist-packages (0.1.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from tensorflow-object-detection-api) (3.2.2)
Requirement already satisfied: Protobuf in /usr/local/lib/python3.7/dist-packages (from tensorflow-object-detection-api) (3.19.4)
Requirement already satisfied: Cython>=0.28.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow-object-detection-api) (0.29.32)
Requirement already satisfied: tensorflow in /usr/local/lib/python3.7/dist-packages (from tensorflow-object-detection-api) (2.9.2)
Requirement already satisfied: twine in /usr/local/lib/python3.7/dist-packages (from tensorflow-object-detection-api) (4.0.1)
Requirement already satisfied: contextlib2 in /usr/local/lib/python3.7/dist-packages (from tensorflow-object-detection-api) (0.5.5)
Requirement already satisfied: wheel in /usr/local/lib/python3.7/dist-packages (from tensorflow-object-detection-api) (0.37.1)
Requirement already satisfied: Pillow>=1.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow-object-detection-api) (7.1.2)
Requirement already satisfied: lxml in /usr/local/lib/python3.7/dist-packages (from tensorflow-object-detection-api) (4.0.1)
Requirement already satisfied: jupyter in /usr/local/lib/python3.7/dist-packages (from tensorflow-object-detection-api) (1.0.0)
```

Anexo 25. Instalar tensorflow-object-detection-api. Información obtenida del sistema desarrollado, Elaborado por el autor.

Se instala el tensorflow-object-detection-api con la lista de requerimientos

```
In [ ]: VERIFICATION_SCRIPT = os.path.join(paths['API_MODEL_PATH'], 'research', 'object_detection', 'builders', 'model_builder_tf2_test.py')
        !python $VERIFICATION_SCRIPT

Running tests under Python 3.7.13: /usr/bin/python3
[ RUN      ] ModelBuilderTF2Test.test_create_center_net_deepmac
2022-09-06 17:26:28.362531: W tensorflow/core/common_runtime/gpu/gpu_bfc_allocator.cc:42] Overriding orig_value setting because the TF_FORCE_GPU_ALLOW_GROWTH environment variable is set. Original config value was 0.
W0906 17:26:29.142760 139938518574976 model_builder.py:1102] Building experimental DeepMAC meta-arch. Some features may be omitted.
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_create_center_net_deepmac): 2.42s
I0906 17:26:29.727083 139938518574976 test_util.py:2459] time(__main__.ModelBuilderTF2Test.test_create_center_net_deepmac): 2.42s
[ OK      ] ModelBuilderTF2Test.test_create_center_net_deepmac
[ RUN      ] ModelBuilderTF2Test.test_create_center_net_model0 (customize_head_params=True)
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_create_center_net_model0 (customize_head_params=True)): 0.96s
I0906 17:26:30.092663 139938518574976 test_util.py:2459] time(__main__.ModelBuilderTF2Test.test_create_center_net_model0 (customize_head_params=True)): 0.96s
[ OK      ] ModelBuilderTF2Test.test_create_center_net_model0 (customize_head_params=True)
[ RUN      ] ModelBuilderTF2Test.test_create_center_net_model1 (customize_head_params=False)
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_create_center_net_model1 (customize_head_params=False)): 0.47s
I0906 17:26:31.163128 139938518574976 test_util.py:2459] time(__main__.ModelBuilderTF2Test.test_create_center_net_model1 (customize_head_params=False)): 0.47s
```

Anexo 26. Verificar las rutas de los archivos. Información obtenida del sistema desarrollado, Elaborado por el autor.

```
In [ ]: import tensorflow as tf
print(tf.__version__)

2.9.2
```

Anexo 27. Importar Tensorflow y se verifica la versión. Información obtenida del sistema desarrollado, Elaborado por el autor.

1.- Se obtiene la ruta del modelo pre entrenado

2.- Acceder a la carpeta y abrimos

3.- Descargar el modelo pre-entrenado

```
In [ ]: if os.name == 'posix':
    wget (PRETRAINED_MODEL_URL)
    #demostramos el archivo
    mv (PRETRAINED_MODEL_NAME+'.tar.gz') {paths['PRETRAINED_MODEL_PATH']}
    cd (paths['PRETRAINED_MODEL_PATH']) && tar -zxvf {PRETRAINED_MODEL_NAME+'.tar.gz'}
if os.name == 'nt':
    wget.download(PRETRAINED_MODEL_URL)
    move (PRETRAINED_MODEL_NAME+'.tar.gz') {paths['PRETRAINED_MODEL_PATH']}
    cd (paths['PRETRAINED_MODEL_PATH']) && tar -zxvf {PRETRAINED_MODEL_NAME+'.tar.gz'}

--2022-09-06 17:27:07-- http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8.tar.gz
Resolving download.tensorflow.org (download.tensorflow.org)... 188.177.13.128, 2607:fb80:400c:c09::80
Connecting to download.tensorflow.org (download.tensorflow.org)|188.177.13.128|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 20515344 (20M) [application/x-tar]
Saving to: 'ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8.tar.gz'

ssd_mobilenet_v2_fpn 100%[=====] 19.56M 78.0MB/s in 0.3s

2022-09-06 17:27:07 (78.0 MB/s) = 'ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8.tar.gz' saved [20515344/20515344]

ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8/
ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8/checkpoint/
ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8/checkpoint/ckpt-0.data-00000-of-00001
ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8/checkpoint/checkpoint
ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8/checkpoint/ckpt-0.index
ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8/pipeline.config
```

Anexo 28. Obtener la ruta del modelo pre-entrenado. Información obtenida del sistema desarrollado, Elaborado por el autor.

```
In [ ]: labels = [{ 'name': 'mascarilla', 'id': 1 }, { 'name': 'guante', 'id': 2 }]

with open(files['LABELMAP'], 'w') as f:
    for label in labels:
        f.write('item { \n')
        f.write('\tname: {}\n'.format(label['name']))
        f.write('\tid: {}\n'.format(label['id']))
        f.write('\n')
```

Anexo 29. Crear el mapa de las clases de mascarilla y guante. Información obtenida del sistema desarrollado, Elaborado por el autor.

```
In [ ]: ARCHIVE_FILES = os.path.join(paths['IMAGE_PATH'], 'archive.tar.gz')
if os.path.exists(ARCHIVE_FILES):
    tar -zxvf (ARCHIVE_FILES)
if not os.path.exists(files['TF_RECORD_SCRIPT']):
    #clonamos y abrimos el repositorio en la ruta correspondiente
    !git clone https://github.com/nicknochnack/GenerateTFRecord (paths['SCRIPTS_PATH'])

content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904105628.jpg
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904105631.jpg
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904105632.jpg
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904105633.jpg
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904105635.jpg
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904105637.jpg
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904105638.jpg
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904105640.jpg
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904105641.jpg
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904105643.jpg
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904105644.jpg
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904105646.jpg
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904105647.jpg
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904105649.jpg
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904112933.jpg
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904112933_01.jpg
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904112934.jpg
content/drive/MyDrive/Tesis/Tensorflow/workspace/images/train/IMG20220904112935.jpg
```

Anexo 30. Abrir el archivo de mapa de etiquetas. Información obtenida del sistema desarrollado, Elaborado por el autor.


```
In [ ]: pipeline_config.model.ssd.num_classes = len(labels)
pipeline_config.train_config.batch_size = 4
pipeline_config.train_config.fine_tune_checkpoint = os.path.join(paths['PRETRAINED_MODEL_PATH'], PRETRAINED_MODEL_NAME, 'checkpoint')
pipeline_config.train_config.fine_tune_checkpoint_type = 'detection'
pipeline_config.train_input_reader.label_map_path = files['LABELMAP']
pipeline_config.train_input_reader.tf_record_input_reader.input_path[:] = [os.path.join(paths['ANNOTATION_PATH'], 'train.record')]
pipeline_config.eval_input_reader[0].label_map_path = files['LABELMAP']
pipeline_config.eval_input_reader[0].tf_record_input_reader.input_path[:] = [os.path.join(paths['ANNOTATION_PATH'], 'test.record')]
```

Anexo 36. Crear una configuración de Pipeline. Información obtenida del sistema desarrollado, Elaborado por el autor.

Pipeline_config: se crea una configuración de la canalización de plantilla para hacer los cambios.

- 1.- Se actualiza el valor de la etiqueta en nuestras imágenes
- 2.- Se pasa a la configuración del lote, por lo que el tamaño del lote del punto de configuración del tren de configuraciones y se lo ha establecido en 4, esto dependerá de la capacidad de hardware
- 3.- Se especifica el punto de control y se descargan modelos pre-entrenados para poder realizar el entrenamiento desde el punto de control checkpoint0
- 4.- Se especificar entrenar un modelo de detección
- 5.- Se coloca la ruta del mapa de etiqueta donde mostraba anteriormente los registros tf.
- 6.- Indicamos que los registros de entrenamiento están dentro de nuestra ruta de anotación
- 7.- Las dos últimas líneas se indica algo referente a los componentes de prueba

```
In [ ]: pipeline_config = pipeline_pb2.TrainEvalPipelineConfig()
#implementaciones para guardar y cargar puntos de control
with tf.io.gfile.GFile(files['PIPELINE_CONFIG'], "r") as f:
    proto_str = f.read()
    text_format.Merge(proto_str, pipeline_config)
```

Anexo 37. Se abren y predicen las 90 clases del modelo pre-entrenado. Información obtenida del sistema desarrollado, Elaborado por el autor.

```
In [ ]: TRAINING_SCRIPT = os.path.join(paths['APIMODEL_PATH'], 'research', 'object_detection', 'model_main_tf2.py')
```

Anexo 38. Convertir el mensaje protobuf en cadena. Información obtenida del sistema desarrollado, Elaborado por el autor.

```
!apt install --allow-change-held-packages libcudnn8=8.1.0.77-1+cuda11.2
```

Anexo 39. Instalar CUDA. Información obtenida del sistema desarrollado, Elaborado por el autor.

```
In [ ]: config_text = text_format.MessageToString(pipeline_config)
with tf.io.gfile.GFile(files['PIPELINE_CONFIG'], "wb") as f:
    f.write(config_text)
```

Anexo 40. Configurar para guardar el modelo en la guía de entrenamiento

```
In [ ]: command = "python {} --model_dir={} --pipeline_config_path={} --num_train_steps=2000".format(TRAINING_SCRIPT, paths['model_dir'], paths['pipeline_config_path'])
print(command)

python /content/drive/MyDrive/Tesis/Tensorflow/models/research/object_detection/model_main_tf2.py --model_dir=/content/drive/MyDrive/Tesis/Tensorflow/workspace/models/my_ssd_mobnet --pipeline_config_path=/content/drive/MyDrive/Tesis/Tensorflow/workspace/models/my_ssd_mobnet/pipeline.config --num_train_steps=2000
```

Anexo 41. Acceder a la ruta del modelo. Información obtenida del sistema desarrollado, Elaborado por el autor.

```
In [ ]: import os
import tensorflow as tf
from object_detection.utils import label_map_util
from object_detection.utils import visualization_utils as viz_utils
from object_detection.builders import model_builder
from object_detection.utils import config_util
```

Anexo 42. Trabajar con los mapas de etiquetas. Información obtenida del sistema desarrollado, Elaborado por el autor.

- 1.- El sistema Operativo
- 2.- Utilidad del mapa de etiquetas: Permitirá trabajar con nuestro mapa de etiquetas
- 3.- Utilidad de visualizaciones: Nos permitirá ver a superposición en nuestra imagen para que realmente veamos los cuadros delimitadores
- 4.- Generador de modelos: Nos permitirá construir nuestro modelo desde nuestro punto de control y nuestro archivo.

```
In [ ]: # Load pipeline config and build a detection model
configs = config_util.get_configs_from_pipeline_file(files['PIPELINE_CONFIG'])
detection_model = model_builder.build(model_config=configs['model'], is_training=False)

# Restore checkpoint
ckpt = tf.compat.v2.train.Checkpoint(model=detection_model)
ckpt.restore(os.path.join(paths['CHECKPOINT_PATH'], 'ckpt-3')).expect_partial()

@tf.function
def detect_fn(image):
    image, shapes = detection_model.preprocess(image)
    prediction_dict = detection_model.predict(image, shapes)
    detections = detection_model.postprocess(prediction_dict, shapes)
    return detections
```

Anexo 43. Configurar el modelo pre-entrenado. Información obtenida del sistema desarrollado, Elaborado por el autor.

- 1.- Se construye la configuración de modelo y especificamos que no está entrenando en este momento para después poder restaurar nuestro punto de control, pasando a través de nuestro modelo de detección tf.compat.v2
- 2.- Restauramos el punto de control con 3 por lo que este fue el último que entrenamos.

```
In [ ]: import cv2
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
```

Anexo 44. Importar CV2, Numpy, Matplotlib. Información obtenida del sistema desarrollado, Elaborado por el autor.

1.- cv2: obtención y reconocimientos de imágenes

2.- Numpy: Crea vectores y matrices

3.- Matplotlib: Genera gráficos a partir de los datos

```
In [ ]: category_index = label_map_util.create_category_index_from_labelmap(files['LABELMAP'])
```

Anexo 45. Categoría para mapear las etiquetas. Información obtenida del sistema desarrollado, Elaborado por el autor.

El índice de categoría se crea para mapear las etiquetas o utilidades pasado a través de nuestra ruta de mapa de etiquetas.

```
In [ ]: !apt install --allow-change-held-packages libcudnn8=8.1.0.77-1+cuda11.2

Reading package lists... Done
Building dependency tree
Reading state information... Done
libcudnn8 is already the newest version (8.1.0.77-1+cuda11.2).
The following package was automatically installed and is no longer required:
  libnvidia-common-460
Use 'apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 18 not upgraded.
```

Anexo 46. Descargar CUDA con la potencia general GPU, de los núcleos de NVIDIA. Información obtenida del sistema desarrollado, Elaborado por el autor.

```
In [ ]: IMAGE_PATH = "/content/drive/MyDrive/Tesis/Tensorflow/workspace/images/test/IMG20220904185656.jpg"

img = cv2.imread(IMAGE_PATH)
image_np = np.array(img)

input_tensor = tf.convert_to_tensor(np.expand_dims(image_np, 0), dtype=tf.float32)
detections = detect_fn(input_tensor)

num_detections = int(detections.pop('num_detections'))
detections = {key: value[0, :num_detections].numpy()
               for key, value in detections.items()}
detections['num_detections'] = num_detections

# detection_classes should be ints.
detections['detection_classes'] = detections['detection_classes'].astype(np.int64)

label_id_offset = 1
image_np_with_detections = image_np.copy()

viz_utils.visualize_boxes_and_labels_on_image_array(
    image_np_with_detections,
    detections['detection_boxes'],
    detections['detection_classes']+label_id_offset,
    detections['detection_scores'],
    category_index,
    use_normalized_coordinates=True,
    max_boxes_to_draw=5,
    category_index,
    use_normalized_coordinates=True,
    max_boxes_to_draw=5,
    min_score_thresh=.7,
    agnostic_mode=False,
    line_thickness=12,)

plt.imshow(cv2.cvtColor(image_np_with_detections, cv2.COLOR_BGR2RGB))
plt.show()
```



Anexo 47. Configuración de la imagen para la detección de los objetos. Información obtenida del sistema desarrollado, Elaborado por el autor.

- 1.- Se captura la imagen para que muestre un cuadro que configura para la detección de los dos objetos
- 2.- Convierte ese cuadro en una matriz, de modo que cuando se pase por la imagen que se desea, esta sea una matriz numpy para poder trabajar con tensorflow
- 3.- Se convierte en un tensorflow para poder pasarlo por un `np.expand_dims`, por lo que se lo colocará dentro de otra matriz porque tensorflow o el modelo espera a que aparezca múltiples imágenes.
- 4.- Usa la función de detección
- 5.- Se define para pasar a un tensorflow de entrada, pasándolo por una gran cantidad de detecciones
- 6.- Se toma la cantidad de detecciones por lo que se va hacer un Pre-Procesamiento de imágenes
- 7.- Especifica una constante para la compensación de identificación de etiqueta de modo que cuando obtenga nuestras detecciones comenzaran en 0
- 8.- Captura la imagen con np, porque al pasar esta función se visualizará en cuadros y etiquetas en la matriz de imágenes
- 9.- Visualiza cuadros y etiquetas en la matriz de imágenes con puntaje en la esquina superior izquierda, pasando por el índice de categoría y coordenadas normalizadas.
- 10.- Configura el umbral con 7, para diferenciar los objetos como mascarilla, guantes, e inactivos
- 11.- Muestra por pantalla con el comando `cv2.imshow`
- 12.- Se colocó líneas de interrupción al cerrar

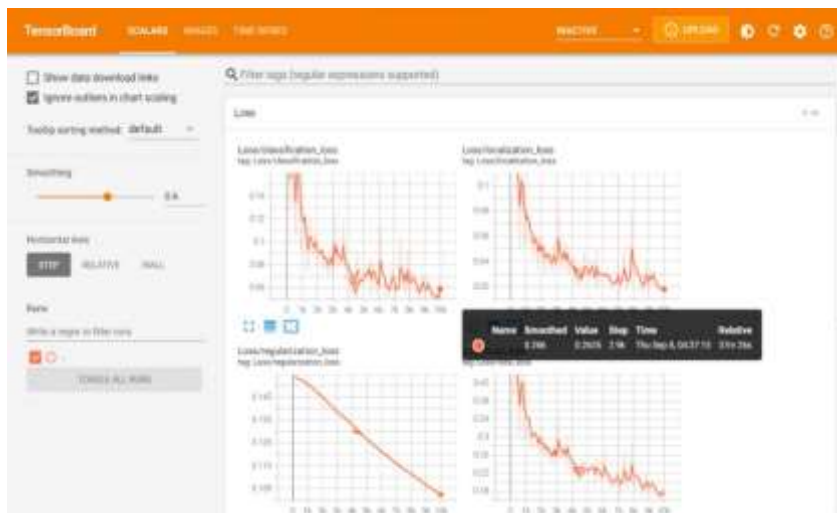
```
# Abrir camara en tiempo real
while cap.isOpened():
    ret, frame = cap.read()
    # 1 corresponde a deteccion de mascarilla
    # 3 corresponde a deteccion de guante
    if (1 in classes_detected) and (3 in classes_detected):
        frame[ y:y+img_height , x:x+img_width ] = img_check
    else:
        frame[ y:y+img_height , x:x+img_width ] = img_alert

    image_np = np.array(frame)

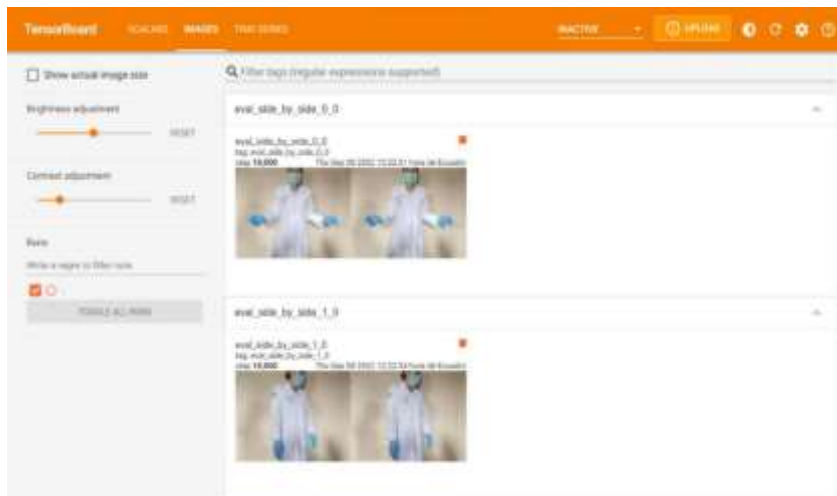
    input_tensor = tf.convert_to_tensor(np.expand_dims(image_np, 0), dtype=tf.float32)
```

Anexo 48. Condiciones según se detecten todos o no de los EPP. Información obtenida del sistema desarrollado, Elaborado por el autor.

Anexos de los resultados del sistema en DashBoard



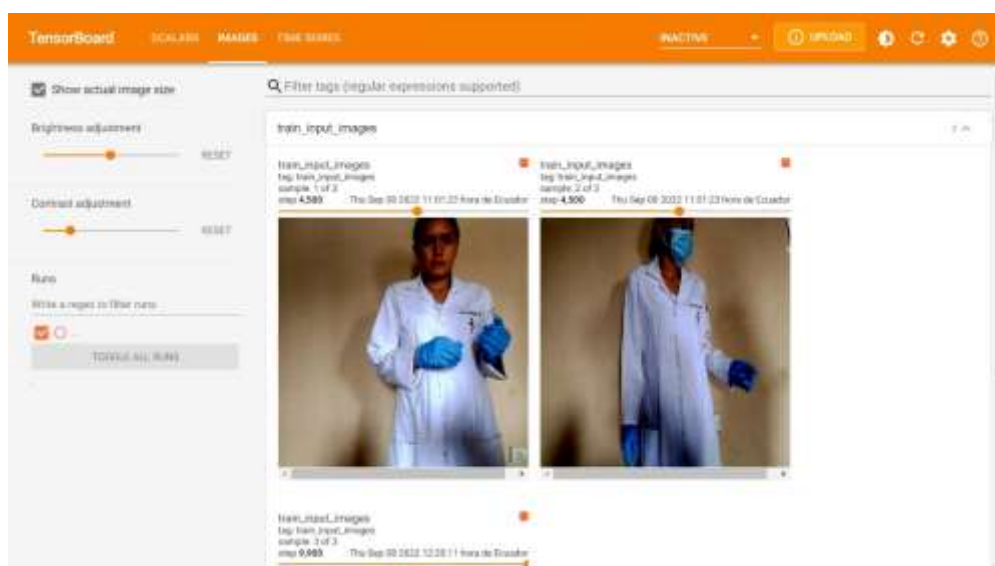
Anexo 49. Resultados del proceso de entrenamiento del sistema en dashboard. Información obtenida del sistema desarrollado, Elaborado por el autor.



Anexo 50. Proceso de entrenamiento del sistema. Información obtenida del sistema desarrollado, Elaborado por el autor.



Anexo 51. Resultados de imágenes de prueba. Información obtenida del sistema desarrollado, Elaborado por el autor.



Anexo 52. Imágenes de la carpeta de la carpeta de train para entrenamiento. Información obtenida del sistema desarrollado, Elaborado por el autor.

Formato de la entrevista



ENTREVISTA A ENCARGADO DEL PERSONAL "VIDASSAN"

Desarrollo de un sistema con inteligencia artificial para el procesamiento de vídeo en la detección del uso de equipos de seguridad personal de los técnicos del laboratorio clínico VIDASSAN de la ciudad de Guayaquil.

La presente entrevista tiene como objetivo obtener información relevante, sobre la situación inicial en el laboratorio clínico VIDASSAN acerca del control del uso de los equipos de protección personal.

[Acceder a Google](#) para guardar el progreso. [Más información](#)

Nombre

Tu respuesta

1. ¿Qué son los equipos de protección personal?

Tu respuesta

2. Podría describirme las consecuencias de la falta del equipo de protección personal dentro del laboratorio

Tu respuesta

3. ¿Existe un reglamento interno sobre el uso de los equipos de protección personal por parte de los técnicos?

4. ¿Existe alguna persona encargada de verificar que los técnicos usen los EPP y con qué frecuencia lo realiza?

Tu respuesta

5. ¿Cuáles son los equipos de protección personal indispensables para que los técnicos trabajen?

6. ¿Ha evidenciado problemas por la falta de responsabilidad por parte de los colaboradores al no usar todos los EPP adecuados?

Tu respuesta

7. ¿Aplican alguna sanción en caso de que los colaboradores no usen todos los EPP?

Tu respuesta

8. ¿Cuáles son los beneficios que se pueden obtener al tener un sistema de vigilancia constante de los EPP?

Tu respuesta

9. ¿Ha escuchado sobre un sistema que permita detectar los EPP a través de la detección de vídeo?

Tu respuesta

10. ¿El laboratorio estaría dispuesto a invertir en una nueva tecnología para garantizar la seguridad de sus trabajadores dentro del mismo?

Anexo 53. Formato de entrevista a encargado del personal, Información obtenida del autor, Elaborado por el autor.

1. ¿Qué son los equipos de protección personal?

1 respuesta

Bueno, estos equipos son los que sirven para cuidar la piel, o los medios por los cuales los trabajadores pueden resultar afectados en su salud física y biológica.

2. Podría describirme las consecuencias de la falta del equipo de protección personal dentro del laboratorio

1 respuesta

Creería que una de las consecuencias más notorias, es que se expone demasiado al trabajador, su piel, y sus órganos más importante

3. ¿Existe un reglamento interno sobre el uso de los equipos de protección personal por parte de los técnicos?

1 respuesta

Sí, si existe y es el que usa normativa interna para trabajar en nuestro laboratorio

4. ¿Existe alguna persona encargada de verificar que los técnicos usen los EPP y con qué frecuencia lo realiza?

1 respuesta

Sí, yo soy el encargado, mayormente lo realizo bastante seguido, diría que con una frecuencia de una hora a otra.

5. ¿Cuáles son los equipos de protección personal indispensables para que los técnicos trabajen?

1 respuesta

Esto varía según el área en el que trabajan los técnicos porque por ejemplo los que hacen los análisis para ellos es obligatorio el uso de los guantes, batas y mascarilla, pero lo que están en tomografía o radiografía para ellos es necesario la mascarilla y en ocasiones los guantes

6. ¿Ha evidenciado problemas por la falta de responsabilidad por parte de los colaboradores al no usar todos los EPP adecuados?

1 respuesta

No, los colaboradores saben que les puede hacer daño no usar el equipamiento bien

Anexo 54. 6 primeras respuestas de entrevista, Información obtenida del autor, Elaborado por el autor.

7. ¿Aplican alguna sanción en caso de que los colaboradores no usen todos los EPP?

1 respuesta

Sí, existe una sanción monetaria

8. ¿Cuáles son los beneficios que se pueden obtener al tener un sistema de vigilancia constante de los EPP?

1 respuesta

Podría ser que se puede evitar que los empleados salgan infectados por algún virus o enfermedades.

9. ¿Ha escuchado sobre un sistema que permita detectar los EPP a través de la detección de video?

1 respuesta

No, no he escuchado, pero sí acerca de sistemas que detectan otras cosas como carros y así

10. ¿El laboratorio estaría dispuesto a invertir en una nueva tecnología para garantizar la seguridad de sus trabajadores dentro del mismo?

1 respuesta

Podría ser, pero tendría que consultarlo

Anexo 55. 4 últimas respuestas de entrevista, Información obtenida del autor, Elaborado por el autor.

Formato de encuestas y análisis de resultados

Desarrollo de un sistema con inteligencia artificial para el procesamiento de vídeo en la detección del uso de equipos de seguridad personal de los técnicos del laboratorio clínico VIDASSAN de la ciudad de guayaquil.

La presente entrevista tiene como objetivo obtener información relevante directamente de los técnicos del laboratorio clínico VIDASSAN acerca del control del uso de los equipos de protección personal (EPP).

[Acceder a Google](#) para guardar el progreso. [Más información](#)

***Obligatorio**

Es consciente de la peligrosidad que corre al exponerse ante algún peligro biológico sin el uso de algún o algunos equipos de seguridad personal. *

- ☐ Siempre
- ☐ Casi siempre
- ☐ A veces
- ☐ Casi nunca
- ☐ Nunca

¿Cuáles son los equipos de protección personal indispensables para laborar dentro del laboratorio? *

- ☐ Guantes
- ☐ Gorro desechable
- ☐ Mascarilla
- ☐ Cubre zapatos desechables
- ☐ Gafas protectoras
- ☐ Bata

¿El laboratorio tiene equipos disponibles de protección personal para todos los colaboradores? *

- ☐ Siempre
- ☐ Casi siempre
- ☐ A veces
- ☐ Casi nunca
- ☐ Nunca

Las batas, guardapolvos, batas de trabajo o uniformes designados para el uso en el laboratorio deben ser usados de forma obligatoria mientras se trabaja dentro del laboratorio *

- ☐ Siempre
- ☐ Casi siempre
- ☐ A veces
- ☐ Casi nunca
- ☐ Nunca

¿Cuál es la frecuencia de la revisión de los EPP dentro del laboratorio? *

- ☐ Solo a la entrada del laboratorio
- ☐ Cada hora
- ☐ Cada 2 o más horas
- ☐ A la entrada del día laboral y después del amuerzo
- ☐ Nunca

Considera que un sistema de vigilancia con inteligencia artificial para la detección de EPP puede reducir riesgos para su salud y consecuentemente para su entorno. *

- ☐ Totalmente de acuerdo
- ☐ De acuerdo
- ☐ Ni de acuerdo ni en desacuerdo
- ☐ En desacuerdo
- ☐ Totalmente en desacuerdo

¿Cómo le gustaría ver la detección del uso de los EPP? *

- ☐ En recuadros que se vean en la pantalla
- ☐ Ver la pantalla pero que no muestre ninguna indicación
- ☐ No ver nada en la pantalla

¿Cómo le gustaría que le avisara si le hace falta alguno de los EPP? *

- ☐ Con un color rojo indicando que le hace falta un EPP
- ☐ Con alertas de sonido
- ☐ Con ambos
- ☐ Con un mensaje en la pantalla
- ☐ Ninguna alerta

Género *

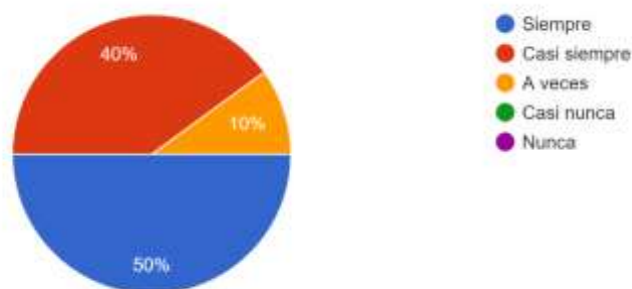
- ☐ Masculino
- ☐ Femenino

Edad *

- ☐ 18 a 25 años
- ☐ 26 a 30 años
- ☐ 31 a 40 años
- ☐ 41 a 49 años
- ☐ Más de 50 años

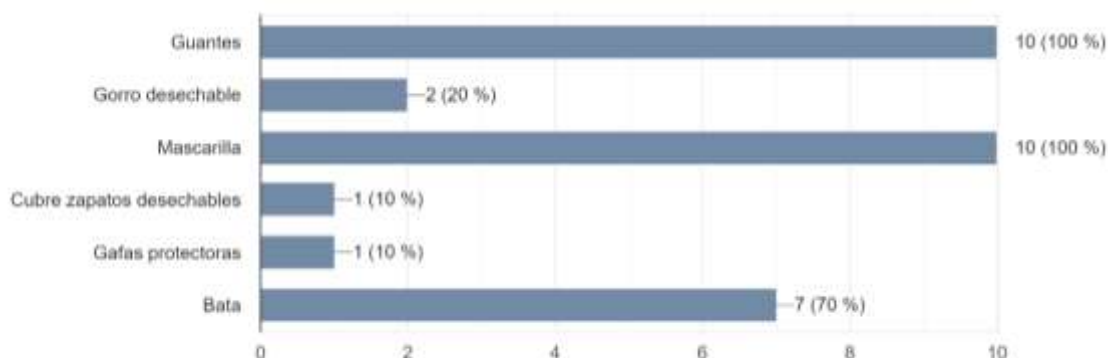
Anexo 56. Formato de las encuestas realizadas. Información obtenida del sistema desarrollado,
Elaborado por el autor.

Resultados de las encuestas



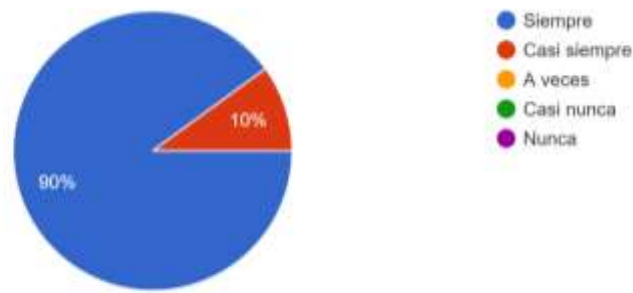
Anexo 57. Diagrama de pastel del conocimiento sobre el peligro al que se exponen. Información obtenida de las encuestas realizadas, Elaborado por el autor.

Análisis: De forma gráfica se puede apreciar que existe un alto conocimiento de parte de los técnicos de laboratorios a cerca del peligro eminente al que se exponen a diario y aún con mayor alerta cuando uno de estos no use el equipo de protección personal, un porcentaje minúsculo a comparación de las respuestas más efectivas, dijo que a veces es consciente de cuán peligro hay en el laboratorio clínico.



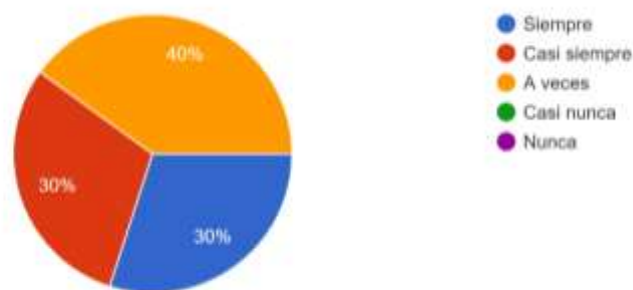
Anexo 58. Diagrama de barras de los EPP más usados por los técnicos. Información obtenida de las encuestas realizadas, Elaborado por el autor.

Análisis: En este gráfico se detallan los tipos de EPP que en su mayoría usan los técnicos de laboratorio, más sin embargo como se presentan sus respuestas, el 100% considera que los EPP más indispensables para laborar son los guantes y la mascarilla, con un 30% menor a la mayoría se encuentra también la bata, con 20% el gorro desechable, y las últimas opciones comparten el 10%



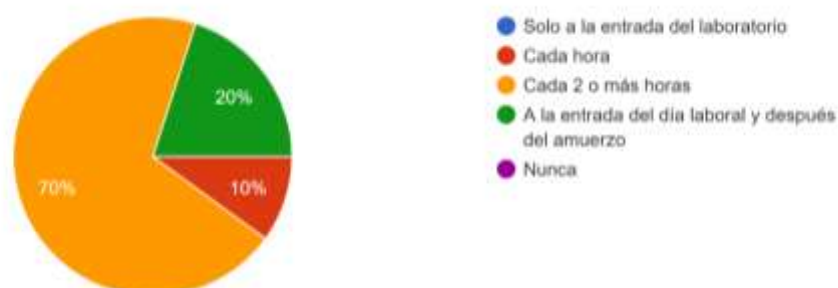
Anexo 59. Diagrama de pastel de la disponibilidad de los EPP. Información obtenida de las encuestas realizadas, Elaborado por el autor.

Análisis: Los resultados de esta incógnita demuestran los intereses del laboratorio por cuidar de la salud de los colaboradores, puesto que siempre representa la mayor opción escogida con un 90%, y un casi siempre con 10%, por lo que se entiende que el laboratorio VIDASSAN provee de forma efectiva estos equipos de seguridad.



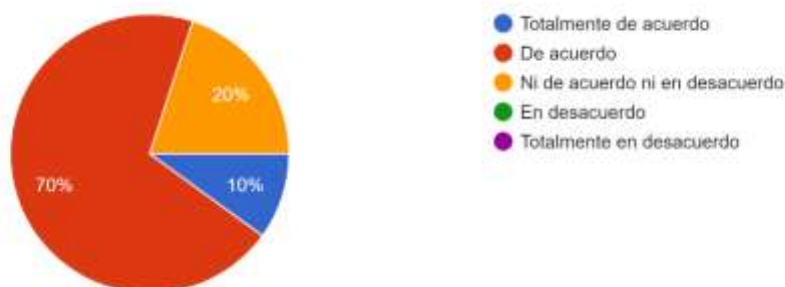
Anexo 60. Diagrama de pastel del porcentaje de importancia de la bata. Información obtenida de las encuestas realizadas, Elaborado por el autor.

Análisis: Se entiende que dentro del laboratorio hay muchos roles de los cuales necesitan el uso de la bata, como por lo contrario que no la necesitan, este último grupo representa la mayoría con 40%, el resto del muestro total está dividido en 30% cada una que son siempre y casi siempre necesitan usar la bata, esto fortaleció la decisión de no detectar los guantes y mascarilla.



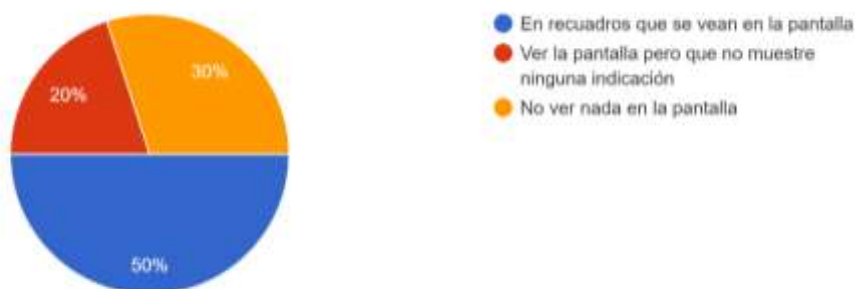
Anexo 61. Diagrama de pastel de la frecuencia de revisión de EPP. Información obtenida de las encuestas realizadas, Elaborado por el autor.

Análisis: Se infiere entonces a través de las respuestas con un 70% que el encargado de los colaboradores del laboratorio verifica que ellos estén usando los EPP correctos con una frecuencia de entre dos o más horas, un 20% indicó en cambio que verifica la indumentaria a la entrada del día laboral y después del almuerzo, también indicaron con un 10% que lo realizaba cada hora.



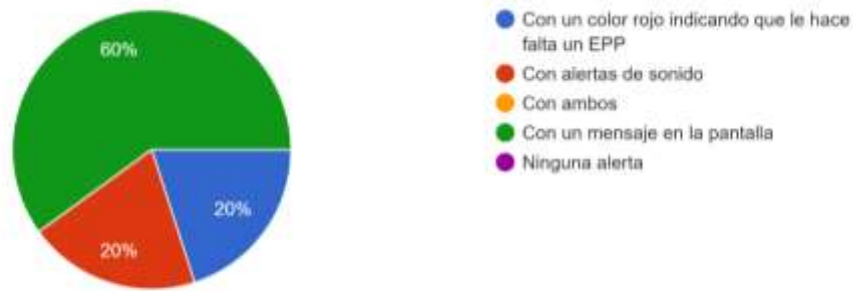
Anexo 62. Diagrama de pastel de la aceptación del sistema de detección de EPP. Información obtenida de las encuestas realizadas, Elaborado por el autor.

Análisis: Se obtuvieron respuestas muy positivas de la aceptación de un sistema de vigilancia con inteligencia artificial para la detección de EPP, un 70% por ciento de los colaboradores está de acuerdo con que este sistema será de utilidad para el laboratorio. Existe un grupo dubitativo que representa el 20%, una de las razones por las cuales no conozca si pueda ser de utilidad este sistema, es por el limitado conocimiento sobre inteligencia artificial, y un 10% indica un total acuerdo.



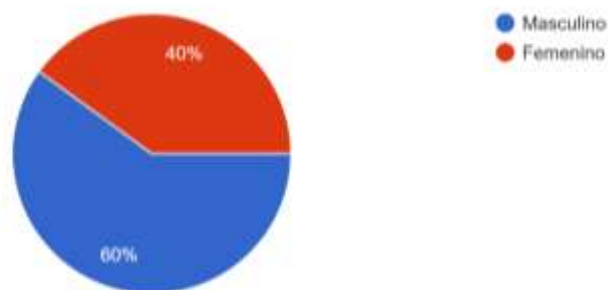
Anexo 63. Diagrama de pastel de cómo se desea visualizar la detección de EPP. Información obtenida de las encuestas realizadas, Elaborado por el autor.

Análisis: La media de los encuestados indica que les gustaría visualizar la identificación de los EPP en recuadros y que los puedan ver en la pantalla, un 30% prefirió no ver ninguna indicación, ni la pantalla, y un 20% desea poder visualizarse en la pantalla pero que no se le indique nada.



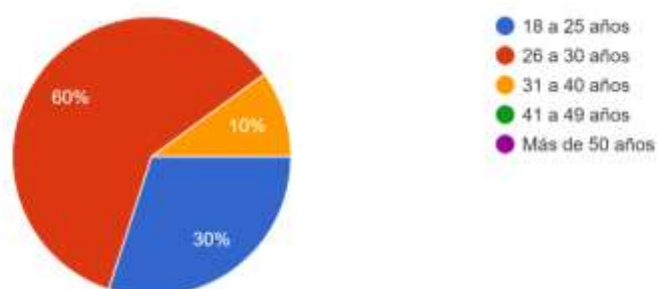
Anexo 64. Diagrama de pastel del tipo de aviso que se desea en el sistema. Información obtenida de las encuestas realizadas, Elaborado por el autor.

Análisis: El tipo de aviso con mayor aceptación que se percibió fue la alerta que se muestra con mensajes en la pantalla con un 60%, el 40% que queda libre, se divide parcialmente entre la opción de poder realizar avisos con un color rojo que indique que le hace falta un EPP, y otros desean alertas de sonido. En general, se observa que se desea una forma de alertarlos, esta pregunta es de gran utilidad para el sistema ya que gracias a esta se decidió alertarlos como la mayoría escogió que es con mensajes en la pantalla.



Anexo 65. Diagrama de pastel del tipo de género de los colaboradores. Información obtenida de las encuestas realizadas, Elaborado por el autor.

Análisis: con este gráfico se determina que el género que más prevalece es el masculino con un 60 %, que representa a 6 de los 10 trabajadores, por lo contrario, el género femenino representa apenas un 40%



Anexo 66. Diagrama de pastel del promedio de la edad de los técnicos. Información obtenida de las encuestas realizadas, Elaborado por el autor.

Análisis: El rango de edad de la mayor parte de los trabajadores del laboratorio VIDASSAN rondan en la edad de 26 a 30 años de edad, considerada como una edad promedio, un 30% están aquellos más jóvenes que van desde los 18 a 25 años, y por último apenas un 10% entre 31 a 40 años.

Fundamentación legal

A continuación, se exponen los diferentes artículos que sustentan este proyecto e intelecto del autor para su desarrollo.

CONSTITUCIÓN POLÍTICA DEL ECUADOR 2008

TÍTULO VI

RÉGIMEN DE DESARROLLO

Capítulo sexto. Trabajo y producción. Sección segunda. Tipos de propiedad.

Art. 322.- Se reconoce la propiedad intelectual de acuerdo con las condiciones que señale la ley. Se prohíbe toda forma de apropiación de conocimientos colectivos, en el ámbito de las ciencias, tecnologías y saberes ancestrales. Se prohíbe también la apropiación sobre los recursos genéticos que contienen la diversidad biológica y la agrobiodiversidad.

TÍTULO VII

RÉGIMEN DEL BUEN VIVIR

Capítulo primero. Inclusión y equidad. Sección octava. Ciencia, tecnología, innovación y saberes ancestrales.

Art. 385.- El sistema nacional de ciencia, tecnología, innovación y saberes ancestrales, en el marco del respeto al ambiente, la naturaleza, la vida, las culturas y la soberanía, tendrá como finalidad:

1. Generar, adaptar y difundir conocimientos científicos y tecnológicos.
2. Recuperar, fortalecer y potenciar los saberes ancestrales.
3. Desarrollar tecnologías e innovaciones que impulsen la producción nacional, eleven la eficiencia y productividad, mejoren la calidad de vida y contribuyan a la realización del buen vivir.

Art. 388.- El Estado destinará los recursos necesarios para la investigación científica, el desarrollo tecnológico, la innovación, la formación científica, la recuperación y desarrollo de saberes ancestrales y la difusión del conocimiento. Un porcentaje de estos recursos se destinará a financiar proyectos mediante fondos concursables. Las organizaciones que 99 reciban fondos públicos estarán sujetas a la rendición de cuentas y al control estatal respectivo.

CÓDIGO ORGÁNICO DE ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN.

Con respecto al uso de tecnologías abiertas para el desarrollo de la detección de EPP en tiempo real, el artículo 142 respalda su uso.

Apartado Segundo

De las tecnologías libres y formatos abiertos.

Artículo 142.-Tecnologías libres

Se entiende por tecnologías libres al software de código abierto, los estándares abiertos, los contenidos libres y el hardware libre. Los tres primeros son considerados como Tecnologías Digitales Libres.

Se entiende por software de código abierto al software en cuya licencia el titular garantiza al usuario el acceso al código fuente y lo faculta a usar dicho software con cualquier propósito. Especialmente otorga a los usuarios, entre otras, las siguientes libertades esenciales:

- La libertad de ejecutar el software para cualquier propósito;
- La libertad de estudiar cómo funciona el software, y modificarlo para adaptarlo a cualquier necesidad. El acceso al código fuente es una condición imprescindible para ello; • La libertad de redistribuir copias; y,
- La libertad de distribuir copias de sus versiones modificadas a terceros.

Se entiende por código fuente, al conjunto de instrucciones escritas en algún lenguaje de programación, diseñadas con el fin de ser leídas y transformadas por alguna herramienta de software en lenguaje de máquina o instrucciones ejecutables en la máquina.

Los estándares abiertos son formas de manejo y almacenamiento de los datos en los que se conoce su estructura y se permite su modificación y acceso no imponiéndose ninguna restricción para su uso. Los datos almacenados en formatos de estándares abiertos no requieren de software propietario para ser utilizados. Estos formatos estándares podrían o no ser aprobados por una entidad internacional de certificación de estándares.

Contenido Libre es el acceso a toda la información asociada al software, incluyendo documentación y demás elementos técnicos diseñados para la entrega necesarios para realizar la configuración, instalación y operación del programa, mismos que deberán presentarse en estándares abiertos.

Se entiende por hardware libre a los diseños de bienes o materiales y demás documentación para la configuración y su respectiva puesta en funcionamiento, otorgan a los usuarios las siguientes libertades otorgan a los usuarios las siguientes libertades:

1. La libertad de estudiar dichas especificaciones, y modificarlas para adaptarlas a cualquier necesidad;
2. La libertad de redistribuir copias de dichas especificaciones; y
3. La libertad de distribuir copias de sus versiones modificadas a terceros.

El Estado en la adquisición de bienes o servicios incluidos los de consultoría de tecnologías digitales, preferirá la adquisición de tecnologías digitales libres. Para el caso de adquisición de software se observará el orden de prelación previsto en este código.

Las resoluciones y ordenanzas expuestas a continuación fueron las que incentivaron la idea de la realización del presente proyecto, para que el mismo sirva como apoyo a las instituciones pertinentes de realizar el control del uso de mascarilla dentro de los buses.

Refiriéndose a la seguridad laboral, en el siguiente reglamento tiene los artículos donde se promueve precautelar al trabajador

REGLAMENTO DEL INSTRUCTIVO ANDINO DE SEGURIDAD Y SALUD EN EL TRABAJO CAPITULO I

GESTION DE LA SEGURIDAD Y SALUD EN EL TRABAJO

Art. 5.- El Servicio de Salud en el Trabajo deberá cumplir con las siguientes funciones:

- a) Elaborar, con la participación efectiva de los trabajadores y empleadores, la propuesta de los programas de seguridad y salud en el trabajo enmarcados en la política empresarial de seguridad y salud en el trabajo;
- b) Proponer el método para la identificación, evaluación y control de los factores de riesgos que puedan afectar a la salud en el lugar de trabajo;
- c) Observar los factores del medio ambiente de trabajo y de las prácticas de trabajo que puedan afectar a la salud de los trabajadores, incluidos los comedores, alojamientos y las instalaciones sanitarias, cuando estas facilidades sean proporcionadas por el empleador;
- d) Asesorar sobre la planificación y la organización del trabajo, incluido el diseño de los lugares de trabajo, sobre la selección, el mantenimiento y el estado de la maquinaria y de los equipos, y sobre las sustancias utilizadas en el trabajo;
- e) Verificar las condiciones de las nuevas Instalaciones, maquinarias y equipos antes de dar inicio a su funcionamiento;
- f) Participar en el desarrollo de programas para el mejoramiento de las prácticas de trabajo, así como en las pruebas y la evaluación de nuevos equipos, en relación con la salud;
- g) Asesorar en materia de salud y seguridad en el trabajo y de ergonomía, así como en materia de equipos de protección individual y colectiva;
- h) Vigilar la salud de los trabajadores en relación con el trabajo que desempeñan;
- i) Fomentar la adaptación al puesto de trabajo y equipos y herramientas, a los trabajadores, según los principios ergonómicos y de bioseguridad, de ser necesario;
- j) Cooperar en pro de la adopción de medidas de rehabilitación profesional y de reinserción laboral;

Art. 11.- El Comité de Seguridad y Salud en el Trabajo tendrá, entre otras, las siguientes funciones:

- a) Participar en la elaboración, aprobación, puesta en práctica y evaluación de las políticas, planes y programas de promoción de la seguridad y salud en el trabajo, de la prevención de accidentes y enfermedades profesionales;
- b) Considerar las circunstancias y colaborar con la investigación de las causas de todos los accidentes, enfermedades profesionales e incidentes que ocurran en el lugar de trabajo;
- c) Hacer recomendaciones pertinentes para evitar la repetición de los accidentes y la ocurrencia de enfermedades profesionales;

- d) Hacer inspecciones periódicas del lugar de trabajo y de sus instalaciones, maquinarias y equipos, a fin de reforzar la gestión preventiva;
- e) Hacer recomendaciones apropiadas para el mejoramiento de las condiciones y el medio ambiente de trabajo, velar porque se lleven a cabo las medidas adoptadas y examinar su eficiencia;
- f) Vigilar el cumplimiento de la legislación, normas internas y las especificaciones técnicas del trabajo relacionadas con la seguridad y salud en el lugar de trabajo;
- g) Procurar el compromiso, colaboración y participación activa de todos los trabajadores en el fomento de la prevención de riesgos en el lugar de trabajo;
- h) Promover que todos los nuevos trabajadores reciban una formación sobre prevención de riesgos, instrucción y orientación adecuada;
- i) Garantizar que todos los trabajadores estén informados y conozcan los reglamentos, instrucciones, especificaciones técnicas de trabajo, avisos y demás materiales escritos o gráficos relativos a la prevención de los riesgos en el lugar de trabajo;
- j) Supervisar los servicios de salud en el trabajo y la asistencia y asesoramiento al empleador y al trabajador;
- k) Conocer los documentos e informes relativos a las condiciones de trabajo que sean necesarios para el cumplimiento de sus funciones, así como los precedentes de la actividad del servicio de prevención, en su caso;
- l) Conocer y aprobar la Memoria y Programación Anual del Servicio de Seguridad y Salud en el Trabajo. Estas funciones deberán desarrollarse de conformidad con la Decisión 584 y la legislación y prácticas de cada País Miembro.

BIBLIOGRAFÍA

- Alarcón Carpio, J. C., & Poma Astete, R. F. (2021) Desarrollo de un algoritmo computacional de detección de equipos de protección eléctrica en personas, orientado a sistemas de vigilancia basados en cámaras IP. [Tesis de ingeniería, Universidad Peruana en Ciencias Aplicadas]. <https://repositorioacademico.upc.edu.pe/handle/10757/657930>
- Alemán Weng Z. (2005). Riesgos en los laboratorios: consideraciones para su prevención. *Revista de Higiene y Sanidad Ambiental*, 5, 132-137. https://saludpublica.ugr.es/sites/dpto/spublica/public/inline-files/bc51015882abc06_Hig.Sanid_.Ambient.5.132-137%282005%29.pdf
- Aranaz Andrés, J., Gea Velázquez de Castro, M., Vicente Guijarro, J., Beltrán Peribáñez, J., García Haro, M., & Valencia-Martín, J. et al. (2020). Mascarillas como equipo de protección individual durante la pandemia de COVID-19: cómo, cuándo y cuáles deben utilizarse. *Journal Of Healthcare Quality Research*, 35(4), 245-252. DOI: 10.1016/j.jhqr.2020.06.001
- Beltrón Macías, F. (2020). Riesgos biológicos en laboratorios clínicos de la ciudad de Portoviejo mediante el método Biogaval. *Revista San Gregorio*, 1(40). 118-130 <https://revista.sangregorio.edu.ec/index.php/REVISTASANGREGORIO/article/view/1418/9-flavio>
- Benavides, A., Pazmiño, D. (2020). Modelo de detección de mascarillas faciales en un sistema embebido en un vehículo autónomo en tiempos de COVID-19. [Tesis de Ingeniería electrónica, Universidad San Francisco De Quito]. <https://repositorio.usfq.edu.ec/bitstream/23000/10125/1/136489.pdf>
- Bobadilla Sancho, J. (2021). Introducción. Gutierrez A (Ed.), *Machine Learning y Deep Learning: Usando Python, Scikit y Keras*. 18-24. <https://books.google.es/books?hl=es&lr=&id=iAAyEAAQBAJ&oi=fnd&pg=PA11&dq=python&ots=QhB4s2sN4s&sig=4fBbnGx7lnuVsrIWS7EyyFNfgXA#v=onepage&q=python&f=false>
- Carmenates Álvarez, B., & Peña Borroto, Y., & Reyes Escobar, A., & Trujillo Pérez, Y., & Rojas Peláez, Y. (2019). La tuberculosis intestinal: un desafío diagnóstico.

Archivo Médico de Camagüey, 23(3), 389-396.

<https://www.redalyc.org/articulo.oa?id=211166531012>

Castelo Pichucho, B. G., & Mosquera López, B. G. (2022). Diseño e implementación de un sistema para la detección y seguimiento de una persona a través de un cuadricóptero de tamaño reducido empleando visión artificial. [Tesis de licenciatura, Escuela Politécnica Nacional]. <https://bibdigital.epn.edu.ec/handle/15000/22126>

Coronel Ortega, F. (2020). Conocimientos sobre transmisión de enfermedades infecciosas durante la práctica clínica, en estudiantes de la facultad de odontología de la universidad nacional del Caaguazú, año 2019. Odontounca. <https://odontounca.edu.py/wp-content/uploads/2021/06/CORONELFABIAN.pdf>

Díaz, F., Toro A. (2020). ARS-CoV-2/COVID-19: el virus, la enfermedad y la pandemia. Revista Medicina & Laboratorio 24(3). 183-199. <https://docs.bvsalud.org/biblioref/2020/05/1096519/covid-19.pdf>

Estrella Gonzalez, María de los Ángeles, & Romero Sacoto, Lilia Azucena, & Ramírez Coronel, Andrés Alexis, & Mesa-Cano, Isabel Cristina (2021). El equipo de protección personal para la atención al paciente con COVID-19: revisión sistemática. Archivos Venezolanos de Farmacología y Terapéutica, 40(7), 666-679. ISSN: 0798-0264. <https://www.redalyc.org/articulo.oa?id=55971545002>

Gur Arie, L. (2022). The practical guide for Object Detection with YOLOv5 algorithm. Towards Data Science. <https://towardsdatascience.com/the-practical-guide-for-object-detection-with-yolov5-algorithm-74c04aac4843>

Herrero Tudela, M. (2021) Clasificación automática de la severidad de la Retinopatía Diabética mediante técnicas de Deep Learning [Tesis de Maestría, Universidad Valladolid]. <https://uvadoc.uva.es/bitstream/handle/10324/50053/TFM-G1502.pdf?sequence=1&isAllowed=y>

Hope T., Yehezkel S. & Lieder I., (2017), Learning Tensorflow: A Guide To Building Deep Learning Systems, Tache N., O'Reilly. 9-21, 51-68

Howard J. & Gugger Sylvain, (2020), Deep Learning for Coders with fastai & PyTorch, Hassell J., Potter M., Faucher C., O'Reilly. 9-33.

- Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., & Weyand, T. et al. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. <https://arxiv.org/abs/1704.04861>
- Iyer, Rakkshab, Bhensdadiya, Kevin & Ringe, Priyansh. (2021). Comparison of YOLOv3, YOLOv5s and MobileNet-SSD V2 for Real-Time Mask Detection. International Journal of Research in Engineering and Technology. International Journal of Research in Engineering and Technology. ResearchGate. 2395-0056. https://www.researchgate.net/publication/353211011_Comparison_of_YOLOv3_YOLOv5s_and_MobileNet-SSD_V2_for_Real-Time_Mask_Detection
- Kanani, Pratik. (2019). Deep Learning to Detect Skin Cancer using Google Colab. International Journal of Engineering and Advanced Technology. 8. 2176-2183. 10.35940/ijeat.F8587.088619.
- Maldonado, M., Duran, C., & Arteaga, J. (2022). Aplicación móvil traductor de señas para la población sordomuda en Colombia implementando técnicas machine learning, talk to hands. Manglar Uninorte. <http://manglar.uninorte.edu.co/handle/10584/10577>
- Manrique Rojas, E (2020). Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo. Revista Ibérica de Sistemas e Tecnologías de Informação, (28), 586-599. <https://www.proquest.com/openview/c7e24c997199215aa26a39107dd2fe98/1?pq-origsite=gscholar&cbl=1006393>
- Massiris, M., Delrieux, C., & Fernández Muñoz, J. Á. (2018). Detección de equipos de protección personal mediante red neuronal convolucional YOLO. In XXXIX Jornadas de Automática. 1022-1029. Área de Ingeniería de Sistemas y Automática, Universidad de Extremadura. DOI capítulo: <https://doi.org/10.17979/spudc.9788497497565.1022> DOI libro: <https://doi.org/10.17979/spudc.9788497497565>
- Moreno, A., & Morillo, C. (2015). Calidad y seguridad en el laboratorio. <https://www.sintesis.com/data/indices/9788490771730.pdf>
- Núñez Huaman, A. (2021) Identificación De Peligros Y Evaluación De Riesgos Para Implementar Medidas De Control En Los Laboratorios Y Talleres De La

- Universidad Nacional De Piura. [Tesis de Ingeniería, Universidad Nacional De Piura]. <https://repositorio.unp.edu.pe/handle/20.500.12676/2966>
- Ochoa Gelvez, E., & Hernández Herrera, G & Trillos Peña, C. (2020). Accidentes laborales por riesgo biológico en trabajadores de laboratorio clínico. Yopal, Colombia. Revista Médica del Instituto Mexicano del Seguro Social, 58(2),144-151. ISSN: 0443-5117. <https://www.redalyc.org/articulo.oa?id=457769376005>
- Ochoa-Gelvez, Edwin Omar, & Hernández-Herrera, Gilma Norella, & Trillos-Peña, Carlos Enrique (2020). Accidentes laborales por riesgo biológico en trabajadores de laboratorio clínico. Yopal, Colombia. Revista Médica del Instituto Mexicano del Seguro Social, 58(2), 144-151. ISSN: 0443-5117. <https://www.redalyc.org/articulo.oa?id=457769376005>
- Orbe Cisneros, E. A., & Vargas Vallejo, V. C. (2022). Diseño e implementación de un sistema de detección de caídas en el adulto mayor mediante visión artificial utilizando redes neuronales convolucionales. [Tesis de Ingeniería, Universidad De Las Fuerzas Armadas]. <http://repositorio.espe.edu.ec/bitstream/21000/31742/1/T-ESPE-052408.pdf>
- Pence, M. (2018). Guide to infection control in the healthcare setting. Sociedad Internacional De Enfermedades Infecciosas. https://isid.org/wp-content/uploads/2019/06/ISID_GUIDE_LABORATORY_AREAS.pdf
- Pérez, J. (2019). Análisis de tráfico vehicular mediante visión artificial. [Tesis de Masterado en automatización y sistemas de control, Universidad técnica de Ambato] https://repositorio.uta.edu.ec/bitstream/123456789/29182/1/Tesis_t1535masc.pdf#page=40&zoom=100,109,289
- Price, M., & Henao, J. (2011). Influencia de la percepción visual en el aprendizaje. Ciencia Y Tecnología Para La Salud Visual Y Ocular, 9(1), 93-101. <https://dialnet.unirioja.es/servlet/articulo?codigo=5599290>
- Quiroz Romero, F. (2020). Mascarillas quirúrgicas a propósito del COVID-19: Algunos aspectos técnicos. Revista Colombiana De Cirugía, 35(2), 200-202. DOI: 10.30944/20117582.620

- Rishma Cahadalawada, S. (2020). Conocimientos Sobre Transmisión De Enfermedades Infecciosas Durante La Práctica Clínica, En Estudiantes De La Facultad De Odontología De La Universidad Nacional Del Caaguazú [Tesis de odontología] <https://odontounca.edu.py/wp-content/uploads/2021/06/CORONEL-FABIAN.pdf>
- Rishma Cahadalawada, S. (2020). Real Time Object Detection and Recognition. [Tesis de Maestría, Blekinge Institute of Technology]. <http://www.diva-portal.org/smash/get/diva2:1414033/FULLTEXT02.pdf>
- Rouhiainen Petteri, L. (2018). Introducción a la inteligencia artificial. Inteligencia artificial 101 cosas que debes saber hoy sobre nuestro futuro. Alienta. 16-32. https://static0planetadelibroscom.cdnstatics.com/libros_contenido_extra/40/39308_Inteligencia_artificial.pdf
- Sahla, N. (2018). A Deep Learning Prediction Model for Object Classification. [Tesis de master, Delft University of Technology]
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. (2019). MobileNetV2: Inverted Residuals and Linear Bottlenecks. <https://arxiv.org/abs/1801.04381>
- Shin, T. (2019). EfficientDet: Scalable and Efficient Object Detection Review. TDS. Towards Data Science. <https://towardsdatascience.com/efficientdet-scalable-and-efficient-object-detection-review-4472ffc34fd9>
- Solawetz, J. (2020). How to Train A Custom Object Detection Model with YOLO v5, Towards Data Science. <https://towardsdatascience.com/how-to-train-a-custom-object-detection-model-with-yolo-v5-917e9ce13208>
- Vieytes S, García K, Numpaque A. (2017). Conocimiento de accidentes de riesgo biológico en estudiantes y trabajadores del área de la salud. Revista Ciencia y Salud Virtual. 9(2). 90-103. DOI:10.22519/21455333.961. <https://revistas.curn.edu.co/index.php/cienciaysalud/article/view/961>