

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Supervisión de acceso del personal a las instalaciones de un ISP,
utilizando redes convolucionales sobre infraestructura Nvidia DGX-A100

PROYECTO DE TITULACIÓN

Previo la obtención del Título de:

Magister en Ciencias de Datos

Presentado por:

Jorge Elieser Pazos Sisalema

Luis Miguel Vaca Arcentales

GUAYAQUIL - ECUADOR

Año: 2021

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Luis Miguel Vaca Arcentales y Jorge Elieser Pazos Sisalema damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”



Luis Miguel Vaca
Arcentales



Jorge Elieser Pazos
Sisalema

COMITÉ EVALUADOR

Ph.D. Andrés Abad Robalino
PROFESOR TUTOR

Ph.D. José Córdova García
PROFESOR EVALUADOR

RESUMEN

Con la ayuda de la visión por computadora, se pueden plantear soluciones a problemas existentes en diversos ámbitos, uno de ellos es el reconocimiento de objetos, el cual puede ser utilizado para detectar el uso de mascarillas y cascos de seguridad que llevan las personas como equipo de protección personal. El presente trabajo implementa una solución que ayuda a un Proveedor de Servicio de Telecomunicaciones a fiscalizar el uso de mascarilla y casco, por parte del personal que ingresa a laborar en las instalaciones. Para ello se implementó la red de comunicaciones y seguridades que comunican diferentes elementos de la solución, como son un computador Nvidia DGX-A100, firewall de seguridad perimetral, y una cámara de video vigilancia ubicada en el sitio de pruebas. Para la detección de objetos se utilizó el algoritmo YOLOv5, mismo que fue entrenado con un conjunto de datos de 4.603 fotografías obtenidas de personal de la empresa y de internet. Se implementó una interfaz gráfica basa en web para el registro de las cámaras y descarga de reportes de cumplimiento de normas de seguridad de la empresa. Se ejecutó las pruebas del sistema durante un mes y se obtuvo un 90.2% de informes correctamente generados.

Palabras Clave: YOLOv5, casco, mascarilla, Nvidia DGX-A100.

ABSTRACT

With the help of computer vision, solutions can be proposed to existing problems in various fields, one of them is object recognition, which can be used to detect the use of masks and safety helmets worn by people as personal protective equipment. The present work implements a solution that helps a Telecommunications Service Provider to control the use of masks and helmets by the personnel that enters to work in the facilities. For this purpose, a communications and security network was implemented to communicate different elements of the solution, such as an Nvidia DGX-A100 computer, a perimeter security firewall, and a video surveillance camera located at the test site. The YOLOv5 algorithm was used for object detection, which was trained with a dataset of 4,603 photographs obtained from company personnel and the Internet. A web-based graphical interface was implemented to register the cameras and download the company's security compliance reports. The system was tested for one month and 90.2% of correctly generated reports were obtained.

Keywords: YOLOv5, helmet, mask, Nvidia DGX-A100.

ÍNDICE GENERAL

COMITÉ EVALUADOR	ii
RESUMEN	iii
<i>ABSTRACT</i>	iv
ÍNDICE GENERAL	v
ABREVIATURAS.....	vii
SIMBOLOGÍA.....	viii
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABLAS	xi
CAPÍTULO 1	1
1. PLANTEAMIENTO DE LA PROBLEMÁTICA	1
1.1 Descripción del problema.....	1
1.2 Justificación.....	2
1.3 Solución Propuesta	3
1.4 Objetivos	4
1.5 Metodología	4
1.6 Resultados Esperados	6
1.7 Dataset.....	6
CAPÍTULO 2	9
2. ESTADO DEL ARTE	9
2.1 Fundamentos del problema	9
2.2 Soluciones de analítica y aprendizaje relacionadas al problema	10
2.3 Fuentes de datos relacionadas al problema	15
2.4 Hardware y software a utilizar	16

CAPÍTULO 3	18
3. DISEÑO E IMPLEMENTACIÓN	18
3.1 Exploración y validación de datos y fuentes	18
3.2 Prototipo del modelo y entrenamiento.....	22
3.3 Infraestructura para procesamiento y almacenamiento	25
3.4 Interfaz gráfica y reportes	30
CAPÍTULO 4	37
4. ANÁLISIS DE RESULTADOS	37
4.1 Métricas y comunicación de resultados	37
4.2 Pruebas de funcionalidad.....	38
4.3 Análisis costo/beneficio	40
5. CONCLUSIONES Y RECOMENDACIONES.....	42
REFERENCIAS.....	43

ABREVIATURAS

CCTV	Circuito Cerrado de TV
COVID-19	Corona virus Disease 19
CUDA	Compute Unified Device Architecture
IA	Inteligencia Artificial
IESS	Instituto Ecuatoriano de Seguridad Social
INEN	Instituto Ecuatoriano de Normalización
ISP	Internet Service Provider
L3 MPLS	Level 3 Multiprotocol Label Switching
MTU	Maximum Transmission Unit
NGFW	Next-Generation Firewall
NTE	Norma tecnológica de la edificación
NTP	Network Time Protocol
OMS	Organización Mundial de la Salud
SGSSO	Sistema de Gestión en Seguridad y Salud Ocupacional
SSDMNV2	Single Shot Multibox Detector and MobileNetV2
UTP	Unshielded Twisted Pair
YOLOv5	You Only Look Once Version 5

SIMBOLOGÍA

dB	decibel
KB	Kilobyte
mAP	mean Average
m	Metro
mV	Milivoltio

ÍNDICE DE FIGURAS

Figura 1.1 Casco de dotación al personal.	6
Figura 1.2 Mascarilla quirúrgica de dotación al personal.	7
Figura 1.3 Personal del ISP con mascarilla y casco.	7
Figura 1.4 Personas utilizando mascarillas y caco. Fuente: Kaggle. [2],[3].....	8
Figura 2.1 Comparación de Yolo V4 en Dataset MS COCO. [7]	11
Figura 2.2 Mosaic Data Augmentation.	12
Figura 2.3 Auto learning bounding box.....	13
Figura 2.4 Cross Stage Partial Networks. [8]	13
Figura 2.5 Nvidia DGX-A100. [13].....	14
Figura 3.1 Muestra de fotografías antes del cleaning.....	18
Figura 3.2 Jupyter Notebook para el proceso de Cleanning.	19
Figura 3.3 Muestra de fotografías luego del cleaning.....	19
Figura 3.4 Etiquetado de casco y mascarilla con LabelImg.	20
Figura 3.5 Muestra de archivo TXT generado con LabelImg.	20
Figura 3.6 Muestra de categorías de imágenes.	21
Figura 3.7 Función de pérdida.	23
Figura 3.8 Precision Vs Recall.	23
Figura 3.9 Métrica mAP (mean Average Precision).	24
Figura 3.10 Puertos configurados en la cámara.....	25
Figura 3.11 Diagrama de Infraestructura.....	26
Figura 3.12 Instalación de cámara en nodo piloto.....	26
Figura 3.13 Parámetros de video.	28
Figura 3.14 Parámetros de red.	29
Figura 3.15 Información de la captura de video.	30

Figura 3.16 Administración de cámaras.	31
Figura 3.17 Formulario de creación de cámara.....	32
Figura 3.18 Seleccionar cámara a modificar.	33
Figura 3.19 Modificar los datos de una cámara.	33
Figura 3.20 Eliminar cámara.	33
Figura 3.21 Generación de reportes.....	34
Figura 3.22 Reporte de acceso al nodo.	35
Figura 3.23 Envío de reporte vía email.....	36
Figura 4.1 Técnico ingresando con mascarilla y casco.	38
Figura 4.2 Técnico ingresando con mascarilla y sin casco.	39
Figura 4.3 Técnico ingresando con casco y sin mascarilla.	39
Figura 4.4 Técnico ingresando sin casco y sin mascarilla.....	40

ÍNDICE DE TABLAS

Tabla 3.1 Fotografías recopiladas.....	20
Tabla 3.2 División del dataset.....	21
Tabla 3.3 Tiempo de entrenamiento.....	22
Tabla 3.4 Especificaciones Huawei Modelo X2221-CL.....	27
Tabla 3.5 Link de conexión a la cámara.....	32
Tabla 4.1 Revisión del departamento de fiscalización vs Sistema de reconocimiento..	37
Tabla 4.2 Accesos al nodo en el mes de septiembre 2021.....	37
Tabla 4.3 Costo mensual en recurso humano.....	40
Tabla 4.4 Costo mensual del proceso de supervisión.....	41
Tabla 4.5 Inversión inicial.....	41

CAPÍTULO 1

1. PLANTEAMIENTO DE LA PROBLEMÁTICA

1.1 Descripción del problema

Para poder comercializar servicios de internet un ISP (Internet Service Provider) debe instalar nodos centrales en todo el territorio donde quiere tener cobertura, los cuales se encuentran interconectados en su mayoría por fibra óptica, estos nodos o cuartos de comunicaciones tienen en su interior racks, equipos de red, tableros eléctricos, climatización, sistemas de protección contra incendios, entre otros, esta infraestructura se debe desplegar en cada ciudad del país. Durante el último año estos cuartos de comunicaciones fueron en promedio 350, este número varia debido a que existe construcción, reubicación y desmontaje de estos.

De estos nodos se despliegan rutas de cableado, las cuales brindan cobertura a una zona determinada y en este recorrido se dispone de cajas de distribución estratégicamente ubicadas de las que se desprenden diferentes tipos de última milla como puede ser fibra óptica, cobre, las cuales cubren manzanas, ciudadelas, barrios de la ciudad, hasta llegar al cliente final con el servicio de internet contratado.

Al realizar las actividades de soporte o activación de nuevos clientes, es necesario que el personal técnico acceda a los nodos, para lo cual se debe hacer un control centralizado de manera manual a todo el personal que ingresa a las instalaciones, debido al gran número de accesos que generan los técnicos, al crecimiento de la red, y teniendo en cuenta que la empresa labora 24x7x365, se necesita personal que haga estas validaciones de seguridad industrial y de bioseguridad, esto se ha vuelto una tarea demandante para el personal de supervisión remota, además ha involucrado la contratación de más recurso humano para cubrir la demanda, y el tiempo de acceso se ha extendido unos minutos.

La actual pandemia ha impuesto nuevas reglas en el trabajo, una de ellas es el cumplimiento de normas de bioseguridad como el uso de mascarillas, según el documento de la OMS publicado el 5 de Junio del 2020 denominado “Recomendaciones sobre el uso de mascarillas en el contexto de la COVID-19” [1], el uso de mascarillas es parte integral de un conjunto de medidas de prevención que pueden evitar la propagación de ciertas enfermedades respiratorias como por ejemplo el COVID-19 , adicional a esto los técnicos deben ingresar con un casco de seguridad, en cumplimiento con el Sistema de Gestión en Seguridad y Salud Ocupacional (SGSSO) como lo establece la Norma OHSAS 18001, al igual que el Modelo Ecuador del SART(Resolución 333 y 390 del IESS), por ende se vuelve una necesidad tener sistemas inteligentes que informen de anomalías en el cumplimiento de las reglas y normas.

1.2 Justificación

Hoy en día existe una constante expansión de la red en el Proveedor de Servicio de telecomunicaciones, lo cual incrementa el número de nodos, por lo que se necesita una supervisión automática del personal de instalación y soporte que ingresan a los cuartos de telecomunicación, debido a que existe un gran número de estos, no se justifica la contratación de personal adicional para realizar la tarea de control de cumplimiento de normas de seguridad y bioseguridad, pues no es una solución escalable.

La supervisión manual de los técnicos que ingresan a los nodos ha extendido el tiempo en el soporte, si bien es cierto el incremento son solo unos minutos, la suma de horas hombre que se pierde es alta, teniendo en cuenta que el personal de soporte en sitio son 2 o 3 técnicos, más el agente de supervisión, genera una demora de 3 minutos adicionales en el ingreso según los informes internos de la empresa. Si multiplicamos este tiempo por el número de personas que intervienen en el proceso, hablamos de entre 6 a 9 minutos que no se aprovecha el recurso humano, generando un impacto económico.

Una consecuencia de lo mencionado anteriormente es la afectación en el tiempo a la cadena de procesos de soporte de la empresa, al final los clientes se ven afectados en un servicio tan indispensable hoy en día como son las telecomunicaciones. Un ejemplo de esto son clientes que trabajan en el sector petrolero, quienes no pueden darse el lujo de paralizar operaciones por falta de comunicación. Por lo tanto, cada minuto que el ISP pueda reducir en la cadena del proceso de soporte, es una gran ventaja para todos los involucrados.

1.3 Solución Propuesta

Para solucionar el problema expuesto anteriormente, se implementará una solución basada en el algoritmo YOLOv5 (You Only Look Once Version 5) mismo que se usa para la detección de objetos en fotografías y videos, este algoritmo se ejecutará en el computador Nvidia DGX-A100 mismo que cuenta con procesadores especiales que trabajan en paralelo y que la empresa ha adquirido para el desarrollo de varios proyectos.

El computador se unirá a un NGFW (Next-Generation Firewall) en donde se realizará las configuraciones necesarias, que permitan únicamente el tráfico requerido, para poder procesar el video en tiempo real proveniente de la cámara instalada en el nodo donde se implementará estas pruebas piloto.

Para que el NGFW pueda comunicarse con la cámara se realizará una conexión lógica de transmisión de datos hasta el nodo, el dispositivo final, en este caso la cámara se conectará con un cable UTP (Unshielded Twisted Pair) certificado.

Adicionalmente se creará una interfaz gráfica que permitirá descargar informes sobre los incumplimientos de estas reglas y normas.

1.4 Objetivos

1.4.1 Objetivo general

Implementar un piloto para la supervisión del personal que ingresa al nodo el cual debe llevar mascarilla y casco como elementos de seguridad.

1.4.2 Objetivos específicos

- Implementar un repositorio de imágenes para el entrenamiento y ajuste del modelo.
- Diseñar una infraestructura segura y escalable para integrar el CCTV (Circuito Cerrado de TV) del nodo en donde se implementará el piloto con el sistema de supervisión y evaluar el modelo de detección de objetos en video.
- Desarrollar una interfaz gráfica que permita la administración de las cámaras, alertas y reportes del cumplimiento de las normas de prevención de riesgos laborales.

1.5 Metodología

- Se realizará el levantamiento de información sobre el equipamiento estándar de protección que usa el personal al momento de ingresar al nodo, concretamente vamos a obtener datos sobre el casco y la mascarilla, para ello realizaremos una reunión con los encargados de verificar el acceso a los nodos y con la Jefatura de los técnicos que ingresan a los nodos. Una vez conocida esta información procederemos a recolectar imágenes que cumplan con las características que nos hayan informado.
- En base a lo mencionado en el punto anterior, se deberá recopilar imágenes de personas utilizando cubrebocas y cascos, esta información será obtenida tanto de los técnicos de la empresa mediante la toma de fotografías, así como de imágenes disponibles que puedan ser descargadas de internet, estas fotografías serán verificadas y etiquetadas manualmente, con lo cual se estandarizara los elementos que forman parte del dataset. Adicional se genera

archivos con las características de cada elemento, este archivo contiene la información de la fotografía como son: Nombre del archivo, tamaño, etiquetas de los elementos, coordenadas de bounding box. Una vez concluido este proceso se guardará las fotos en un servidor interno de la empresa.

- Se deberá implementar las comunicaciones y seguridades a nivel de red, que permitan integrar la infraestructura Nvidia DGX-A100 con la cámara del nodo piloto, a través de canales de datos L3 MPLS, para ello realizaremos las configuraciones necesarias sobre los equipos de red y políticas de seguridad en el firewall, hasta llegar al punto donde se va a instalar la cámara del nodo. En la cámara se configurará los parámetros de red necesarios para la comunicación, y se realizará pruebas de conectividad hacia la infraestructura Nvidia. Esto permitirá integrar de manera segura el origen de datos con el servidor donde se ejecutará el modelo de deep learning.
- Se seleccionará un modelo de detección de objetos que se ajuste a las necesidades del proyecto, para ello nos basaremos en la información científica disponible, en donde se evidencie las ventajas y desventajas de modelos que existen actualmente. Una vez seleccionado el modelo, se realizará el entrenamiento en la nube mediante una cuenta de pago en Google Colab Pro, para ello se deberá utilizar el repositorio de imágenes estandarizadas que se detalló en los pasos previos.
- Una vez entrenado el modelo, se realizará pruebas de toda la infraestructura en su conjunto. Para ello se pondrá en ejecución el modelo sobre el DGX-A100 y simularemos el ingreso del personal a los nodos, proponiendo ciertos escenarios que se pudieran dar en el incumplimiento de las reglas y normas, así como el correcto uso de los elementos de protección personal asignados a cada técnico.
- Se implementará una interfaz gráfica que permita administrar las cámaras instaladas en los nodos, esto quiere decir que se va a ingresar los datos de la cámara y enlazar al nodo al que pertenece. Esta interfaz gráfica deberá permitir agregar, modificar y eliminar cámaras asociadas a los nodos. Adicionalmente deberá generar notificaciones de alerta al operador que se

encuentra en oficinas. También contará con una sección en donde se podrá descargar reportes sobre el incumplimiento de reglas y normas.

1.6 Resultados Esperados

- Tener un repositorio de imágenes recopiladas e imágenes etiquetadas.
- Diagrama de la solución en funcionamiento para la prueba piloto propuesta.
- Generar una gráfica comparativa de resultados de entrenamiento.
- Poner en producción el sistema de detección en tiempo real en donde muestre el “bounding box” en el video de salida.
- Interfaz gráfica de administración de cámaras y visualización de reportes.

1.7 Dataset

Luego de la reunión mantenida con el área de seguridad laboral de la empresa, se ha determinado que los cascos de dotación para los técnicos cumplen con la norma NTE INEN 146 y son mayormente de color azul, aunque también existe de otros colores como amarillo, blanco, rojo, tomate y verde.



Figura 1.1 Casco de dotación al personal.

Adicionalmente las mascarillas provistas son quirúrgicas, de color celeste.

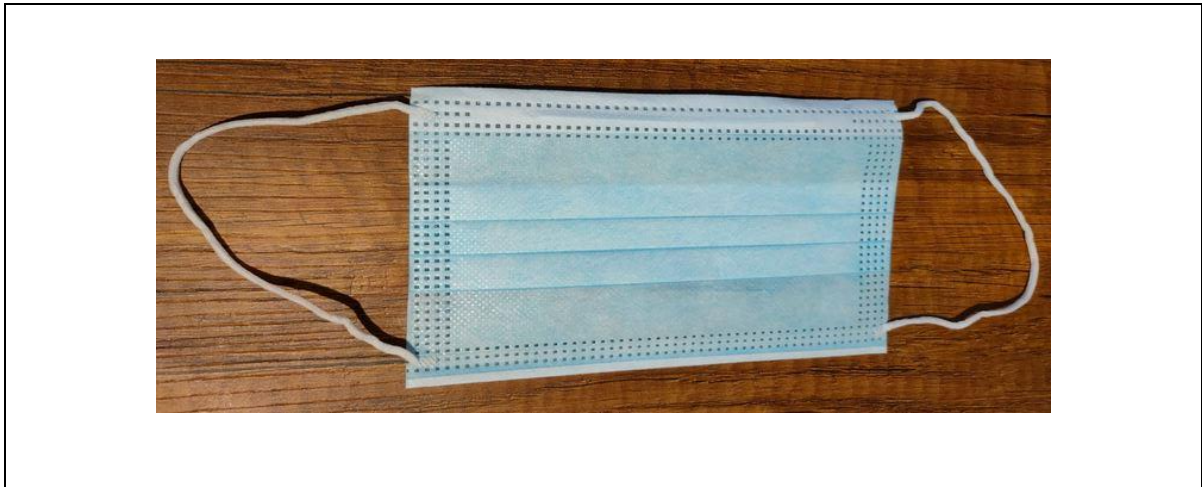


Figura 1.2 Mascarilla quirúrgica de dotación al personal.

El conjunto de datos está compuesto por imágenes de autoría propia, las cuales se recolectaron durante las jornadas laborales de los técnicos del ISP, en estas fotografías el personal se encuentra vistiendo el equipamiento de dotación proporcionado por la empresa, las fotos fueron tomadas desde distintos ángulos.

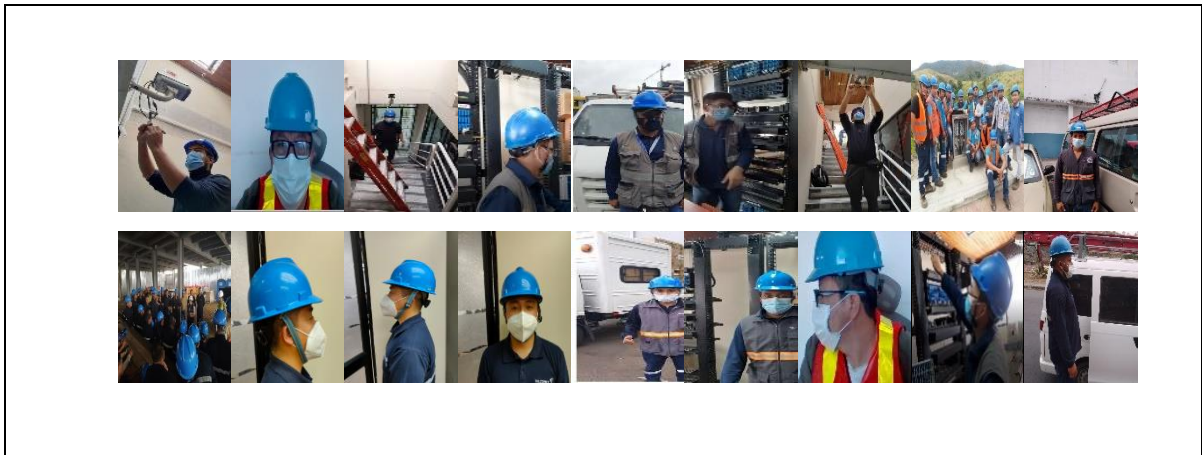


Figura 1.3 Personal del ISP con mascarilla y casco.

Adicionalmente, se utilizó otras fuentes como son repositorios Kaggle, en los cuales se cuenta con fotografías de personas utilizando mascarillas y cascos, a continuación, se presenta una muestra de las imágenes descargadas de internet.

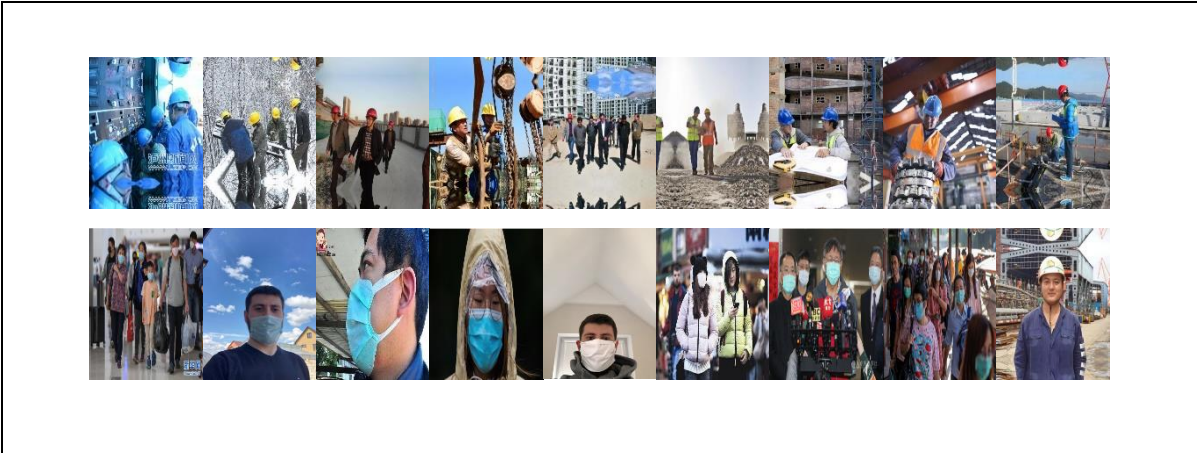


Figura 1.4 Personas utilizando mascarillas y caco. Fuente: Kaggle. [2],[3]

Con todas estas imágenes de las diferentes fuentes, se pudo disponer de un dataset de 4.603 fotografías. En el análisis exploratorio se analizarán las diferentes categorías que constan en el conjunto de datos.

CAPÍTULO 2

2. ESTADO DEL ARTE

2.1 Fundamentos del problema

En el área de las telecomunicaciones existe un conjunto de preocupaciones en la operación de campo, llámese instalación de nueva infraestructura o soporte en sitio, una de ellas es sin duda la seguridad física de los empleados, más aún en estos tiempos donde la pandemia ha obligado a las autoridades de cada país a cambiar ciertas reglas de comportamiento de las personas tanto en espacios públicos como privados, decretando que accesorios de protección se deben utilizar, distanciamiento social, entre otras medidas.

El surgimiento de nuevas tecnologías ha llevado a que las empresas comiencen a poner su atención en las ventajas que pueden ofrecer, generando una necesidad de implementar nuevas herramientas que minimicen los costos operativos. El problema con el que se quiere trabajar es verificar el cumplimiento del uso de mascarillas y cascos al ingresar a los entornos donde la empresa así lo requiera, utilizando visión por computadora.

Actualmente el ISP cuenta con un departamento de verificación manual del cumplimiento de reglas y normas para acceso a los nodos, para cubrir la demanda que exige la operación de la empresa 24x7, se cuenta con recurso humano trabajando en turnos rotativos. A raíz de esta problemática, se ha tratado de investigar una solución tecnológica que se adapte a la infraestructura y necesidades de la empresa.

2.2 Soluciones de analítica y aprendizaje relacionadas al problema

Existen en el mercado dispositivos comerciales como los que ofrece la empresa ZK Tecno con su producto SpeedFace-V5L TD Facial [4], la cual se basa en la detección de mascarillas a una distancia máxima de 2.5 metros, con velocidad de detección de aproximadamente 1 segundo, con reconocimiento facial wide angle de 30 grados, el costo aproximado ronda los mil dólares. Si bien es cierto esta solución de tipo comercial detecta mascarillas, es más costosa por tener varios componentes embebidos dentro del sistema, como una pantalla, medidor de temperatura corporal, mismos que no son requeridos para cumplir los objetivos del proyecto y resta flexibilidad a la hora de integración con la infraestructura diseñada.

En este mismo nicho de mercado también se encuentra presente la empresa hikvision con cámaras AcuSense [5], ideal para locales pequeños, alertas en tiempo real, integración con NVR opcional, detección de mascarilla a distancia máxima de 2 metros, precio promedio de la cámara 250 dólares. Esta compañía también ofrece una solución de análisis de datos en la nube, no contempla la personalización de reportes, tampoco la flexibilidad de integración con los sistemas del ISP. Además, incorpora componentes que no son necesarios para la solución de esta problemática, como son luz estroboscópica y las alarmas de audio.

En la actualidad existe un conjunto de conocimientos dentro de las disciplinas de ciencias de datos, que podemos aprovechar para diseñar una solución que nos ayude a resolver la problemática planteada, en el campo de la visión por computadora se han desarrollado proyectos en diversos ámbitos, como en el sector de la salud para diagnósticos de radiografías, en seguridad utilizando reconocimiento facial en aeropuertos, en sector automotriz para vehículos autónomos, en agricultura para vigilar las plantaciones, entre otros. Todas estas aplicaciones tienen su fundamento en redes neuronales que se han ido desarrollando en los últimos años, al tratarse de la detección de objetos en tiempo real, es necesario pensar que se deban cumplir con algunos aspectos como son velocidad y buena precisión.

Una de las redes que tiene gran desarrollo es YOLO, en diferentes estudios y publicaciones se muestra sus ventajas, características y comparaciones con otros modelos, esta red sale a la luz en 2016 por J Redmon, en su publicación [6] muestra los detalles de esta red convolucional, a diferencia de otras redes de detección de objetos que necesitan que la imagen pase por su red una y otra vez, YOLO solo requiere que este trayecto sea ejecutado una sola vez, de ahí su nombre. En años posteriores el mismo autor desarrolla las versiones 2 y 3, teniendo como una de las limitantes principales la detección de objetos pequeños.

En la publicación de YOLOv4 en abril del 2020 [7], se describe una mejora en términos de velocidad y precisión, adicionalmente se realiza una comparación con las versiones anteriores, así como introduce el tema de aumentación de data al hacer una mezcla de 4 imágenes en una para el entrenamiento. Realiza sus pruebas y comparaciones en el conocido dataset MS COCO, en la siguiente grafica se muestra un detalle sobre el rendimiento y la velocidad en términos FPS (frames per second).

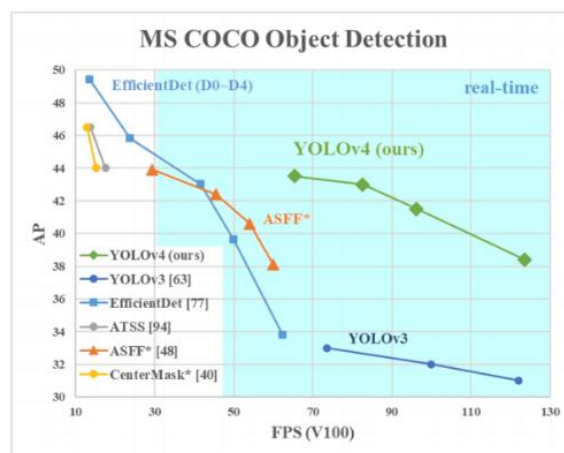


Figura 2.1 Comparación de Yolo V4 en Dataset MS COCO. [7]

Existe gran controversia por YOLOv5, la cual sale a luz en junio del mismo año que la versión 4, no solo por la cercanía de publicaciones, sino también porque la versión 5 no tiene asociada ninguna publicación científica. Una diferencia importante para destacar es que YOLOv5 se implementó sobre Pytorch, un framework que tiene mucha interacción en desarrollo de nuevas aplicaciones y las versiones anteriores están fundamentadas en Darknet.

En YOLOv5 tenemos características importantes como: Mosaic Data Augmentation, Auto learning bounding box y CSP Backbone, esto elementos aumentan el rendimiento en el entrenamiento del modelo. A continuación, se describen cada una de estas características mencionadas.

- **Mosaic Data Augmentation:** A nivel de entrenamiento de un modelo cuanto más data y de diferente clase se disponga mejora el nivel de rendimiento, por lo cual esta función cumple con este objetivo, que es generar un mosaico realizando una mezcla de cuatro imágenes originales. A pesar de que esta función ya existía en la versión 4, su mejora contribuye en disminuir el tiempo de entrenamiento del modelo, a continuación, una muestra de esta característica en nuestro Dataset.

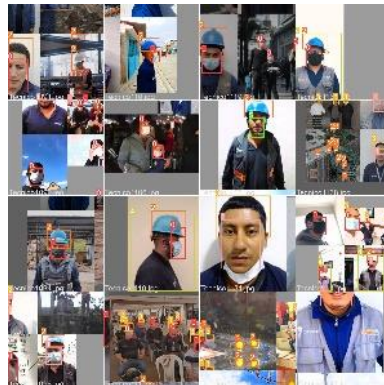


Figura 2.2 Mosaic Data Augmentation.

- **Auto learning bounding box:** El marcar correctamente el objeto a detectar es parte importante del proceso de detección de objetos, adicional a esto consideramos que los objetos pueden ser de diferente tamaño, lo cual influye directamente en el desempeño. El basarse en los Bounding Boxes utilizados en la fase de entrenamiento favorece al desarrollo en la ejecución del modelo.

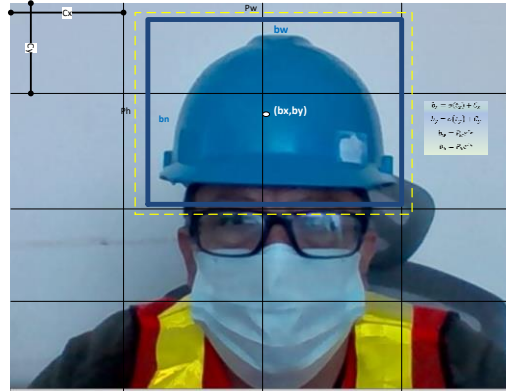


Figura 2.3 Auto learning bounding box.

- **Cross Stage Partial Networks:** el utilizar conexiones recurrentes entre las capas adyacentes no consecutivas ayuda a disminuir el problema de desvanecimiento del gradiente, este detalle fue usado en DenseNet para superar este inconveniente y mejorar su desempeño.

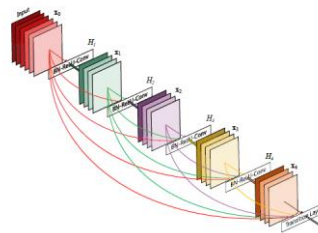


Figura 2.4 Cross Stage Partial Networks. [8]

Un aspecto importante para considerar en la detección de objetos en tiempo real es el hardware, para mejorar el rendimiento en las aplicaciones que se han desarrollado en estos últimos años, los fabricantes han diseñado diversas arquitecturas. Una compañía que ha sido pionera y que ha conseguido mayor impacto en el campo de visión por computadora es Nvidia, pues ha trabajado en múltiples arquitecturas de GPU. Uno de los productos que esta empresa comercializa es el Nvidia DGX-A100, un sistema universal de trabajo con IA, que ofrece una densidad de cálculo sin

precedentes, convirtiéndolo en el primer sistema de IA de 5 petaFLOPS del mundo.[9]. En la figura 2.3 se detallan algunas de sus características técnicas.



Figura 2.5 Nvidia DGX-A100. [13]

En el 2006 se presenta un nuevo producto al mercado llamado CUDA (Compute Unified Device Architecture) [10], esta arquitectura de propósito general permite usar todo el potencial del procesamiento en paralelo de las GPU (graphics processing unit), permitiendo que los desarrolladores puedan acelerar drásticamente sus aplicaciones.

Para realizar el entrenamiento del modelo YOLOv5, se debe realizar un proceso llamado etiquetado de imágenes, que consiste en marcar de manera manual el área donde se encuentra el objeto a detectar dentro de una imagen, luego de esta acción se genera un archivo que contiene la información de las coordenadas del cuadro marcado con su categoría, existen diferentes formatos para este archivo dependiendo de lo que el modelo de detección acepte como entrada. Existe en el mercado diversas herramientas que nos ayudan con este proceso, a continuación, citamos 2 ejemplos.

SuperAnnotate [11], esta plataforma de anotación basada en IA utiliza aprendizaje por transferencia para el proceso de etiquetado de datos. Posee herramientas de anotación de videos e imágenes, una de sus ventajas es marcar los datos de manera rápida con modelos predictivos integrados, por lo que agiliza el proceso de creación de conjunto de datos que facilita la tarea de detección de objetos. Debido a que es

un dataset combinado entre imágenes de internet y propia de la empresa, se ha optado por hacer un etiquetado de manera manual y local, para no exponer las fotografías del personal interno en la nube.

Labellmg [12] es una herramienta gratuita de código abierto que se utiliza para etiquetar imágenes de manera gráfica, se desarrolló en Python y usa QT para su interfaz gráfica. Es muy intuitiva y fácil usar, además de que puede importar y exportar sus anotaciones en varios formatos como son: PASCAL VOC, YOLO y CreateML. Como desventaja podemos mencionar que solo puede crear cuadros rectangulares para marcar los objetos a detectar.

2.3 Fuentes de datos relacionadas al problema

En el sector de la construcción existe un estudio de detección de equipos de protección personal [13], este trabajo abarca el monitoreo y uso adecuado de estos elementos en trabajadores de la construcción. Se utilizó YOLOv2 que es una red neuronal convolucional (CNN), trabaja con Keras sobre Tensorflow y OpenCV. Se puso a prueba el modelo en diversos escenarios y se comprobó su robustez y eficiencia. Cabe mencionar que a la fecha del estudio ya estaba disponible la versión 3 de YOLO, que mejora algunas características de la versión anterior.

Para la detección de mascarillas con SSDMNv2 (Single Shot Multibox Detector and MobileNetV2) detallado en el paper de NAGRATH et al. [14], utilizan una estructura basada en un modelo que combina Resnet50, árboles de decisión, Support Vector Machine (SVM). Para clasificar imágenes se utiliza keras sobre TensorFlow, MobileNetv2, y OpenCV. La solución propuesta permite detectar rostros de manera precisa en diferentes orientaciones, utilizando como base Resnet-10 y OpenCV.

En un estudio realizado por J. leamsaard et al. [15], se trata de encontrar un método eficaz para para la detección de mascarillas, en donde se ha seleccionado como modelo a YOLOv5, el cual es entrenado con 5 diferentes números de épocas 20, 50, 100, 300 y 500. En lo resultados experimentales observaron que el modelo con mayor

rendimiento y precisión fue el creado con 300 épocas, incluso superando al de 500 épocas.

2.4 Hardware y software a utilizar

Dentro del equipo Nvidia DGX-A100 se encuentra corriendo el sistema operativo Ubuntu 20.04, que sirve como base para utilizar Docker [16], que es un proyecto de código abierto que sirve para desplegar aplicaciones dentro de contenedores de software, esto permite asignar recursos y tener separados los proyectos en ejecución. Dentro del contenedor instalaremos varias herramientas de software que nos permitirán cumplir con el objetivo de detección de objetos como son el lenguaje Python [17] el cual es robusto, escalable y de código abierto, actualmente muy usado en proyectos de inteligencia artificial. Utilizaremos la librería Pytorch que es una biblioteca de aprendizaje automático de código abierto, y ejecutaremos el modelo de YOLOv5. Para la herramienta de administración de cámaras y reportaría, usaremos la base de datos MariaDB, Flask y HTML. En el etiquetado de las imágenes utilizamos LabelImg[18] que es una herramienta de código abierto basada en Python, esta última no fue incluida dentro del contenedor.

- **Software:**

- **Ubuntu:** sistema operativo que se ejecuta en el computador Nvidia DGX-A100.
- **Docker:** es un proyecto de código abierto que se utiliza para desplegar aplicaciones dentro de contenedores de software [16], el equipo Nvidia DGX-A100 utiliza esta tecnología para asignar recursos y tener separados los proyectos en ejecución.

- **Python:** lenguaje robusto, escalable y de código abierto, actualmente muy usado en proyectos de inteligencia artificial. [17]
 - **Flask:** Framework escrito en Python y que facilita el desarrollo de Aplicaciones Web bajo el patrón MVC (Modelo Vista Controlador), esto quiere decir que separa los datos que se muestran en la aplicación, la vista o interfaz de usuario (página HTML), y el controlador que gestiona las peticiones web.[19]
 - **Pytorch:** biblioteca de aprendizaje automático de código abierto.
- **YOLOv5:** algoritmo de código abierto para la detección de objetos.
- **MariaDB:** Una base de datos SQL fiable y escalable.[20]
- **LabelImg:** Una herramienta de código abierto basada en Python, bastante usada en etiquetamiento de las imágenes. [18]
- **Hardware:**
 - **Cámara de red marca Huawei Modelo X2221-CL:** Cámara IP para captura de Video en tiempo real.
 - **Nvidia DGX-A100:** sistema universal de trabajo con IA, que ofrece una densidad de cálculo sin precedentes, convirtiéndolo en el primer sistema de IA de 5 petaFLOPS del mundo.[9]
 - **NG Firewall:** Seguridad del tráfico de red entre componentes del proyecto.

CAPÍTULO 3

3. DISEÑO E IMPLEMENTACIÓN

3.1 Exploración y validación de datos y fuentes

Como se describió en el capítulo 1, el dataset está compuesto por imágenes de autoría propia que son las fotografías del personal que labora en la empresa y de fotografías recopiladas de internet. En la Figura 3.1 se observa algunas fotografías antes del proceso de cleaning.

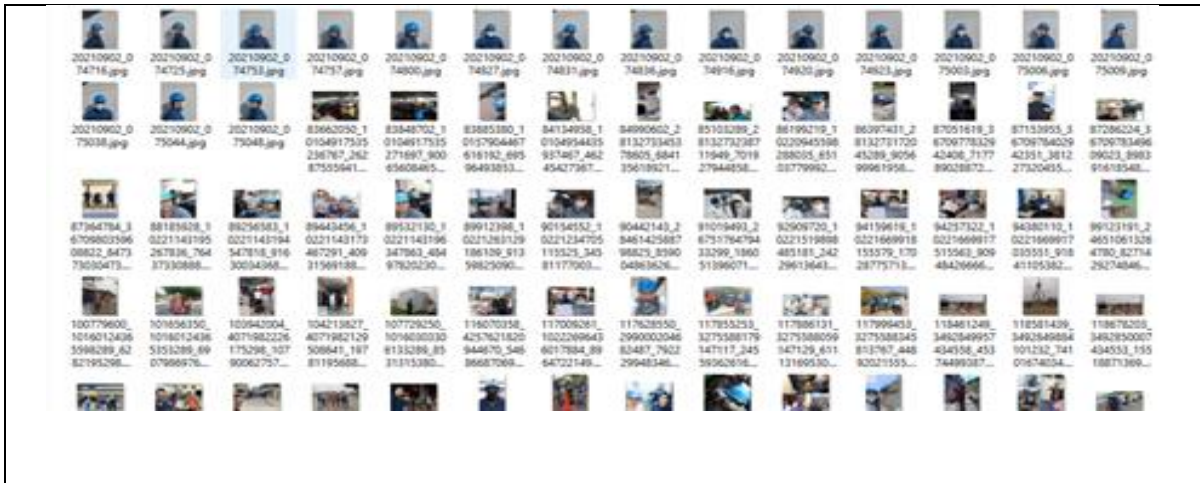


Figura 3.1 Muestra de fotografías antes del cleaning.

Todas las imágenes pasaron por un proceso de cleaning, en la exploración del dataset se inició con la verificación y eliminación de imágenes que no contenían información útil, como por ejemplo imágenes que no mostraban a ninguna persona, fotografías totalmente en negro, etc. Se realizó el proceso de estandarización a nivel de su nombre y extensión del archivo, procesamos 4.603 fotografías, las cuales se comprimieron en un rango de 12MB hasta 300KB.

Usamos las librerías OpenCV y OS de Python, para las tareas de leer, comprimir y reescribir los archivos. En la figura 3.2 se muestra una captura de pantalla del proceso.

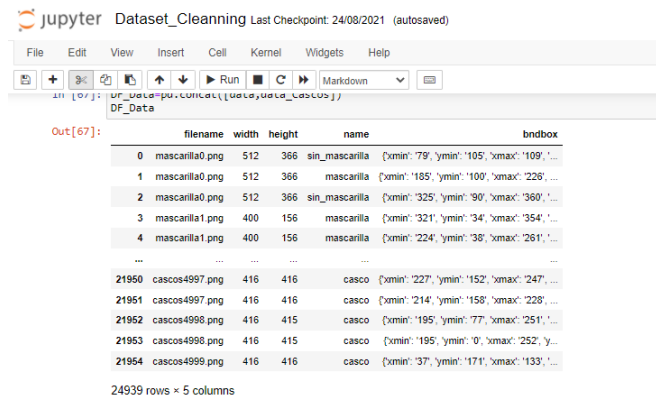


Figura 3.2 Jupyter Notebook para el proceso de Cleaning.

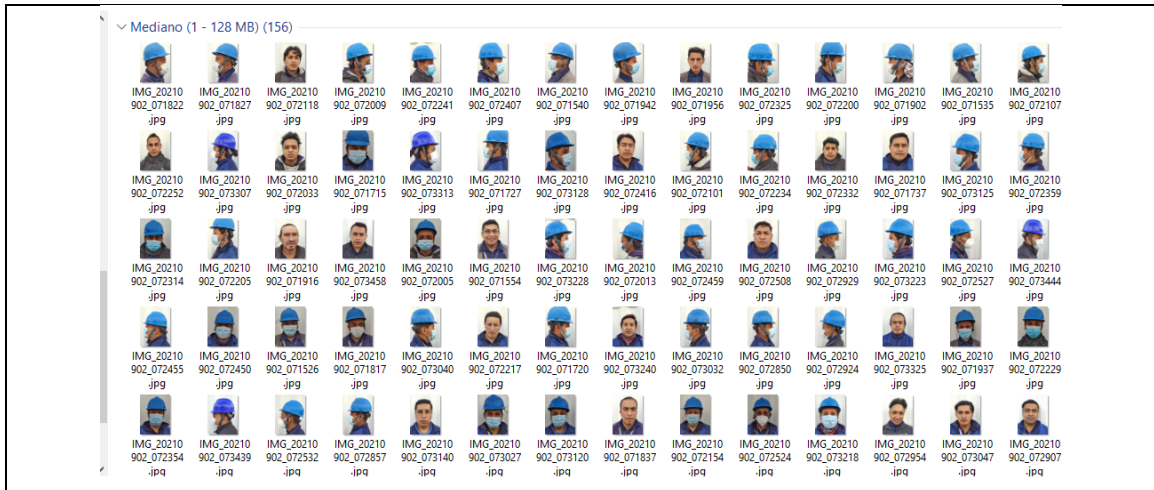


Figura 3.3 Muestra de fotografías luego del cleaning.

Una vez estandarizado el dataset recopilado de las diferentes fuentes, se inició el marcado de los elementos a identificar en cada una de las imágenes como se muestra en la figura 3.4. Esta tarea de etiquetado se ejecutó mediante la

herramienta Labellmg [21], la cual genera un archivo de extensión TXT con las coordenadas del Bounding Box para cada clase.



Figura 3.4 Etiquetado de casco y mascarilla con Labellmg.

En la figura 3.5, se muestra un ejemplo del contenido del archivo generado por Labellmg.

```

cascos43.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
0.3617788596311584 0.359375013387762 0.06971154105849564 0.09375000349245965
2 0.9110577262472361 0.3377403971971944 0.09615384973585606 0.11778846592642367
2 0.7596154129132628 0.41586540010757744 0.12500000465661287 0.1490384670905769
3 0.5192307885736227 0.3365384740754962 0.07692307978868484 0.10576923470944166
2 0.6153846383094788 0.3737980908481404 0.05288461735472083 0.06971154105849564
2 0.9122596493689343 0.025240385555662215 0.09855769597925246 0.05048077111132443
3 0.5204327116953209 0.022836539312265813 0.07932692603208125 0.04567307862453163
    
```

Figura 3.5 Muestra de archivo TXT generado con Labellmg.

En la tabla 3.1 se detalla el número de fotografías recopiladas para el dataset.

Tabla 3.1 Fotografías recopiladas.

CLASE	NÚMERO DE FOTOS
Con mascarilla	1205
Sin mascarilla	803
Con casco	1325
Sin casco	1271
TOTAL	4603

A continuación, hemos seleccionado ejemplos tomados al azar, que representan las categorías mencionadas. Ver figura 3.6.

CATEGORÍAS	FUENTE DE DATOS
Con mascarilla	
Sin mascarillas	
Con casco	
Sin casco	

Figura 3.6 Muestra de categorías de imágenes.

El conjunto de datos se dividió aleatoriamente en 3, entrenamiento 80%, validación 10% y pruebas 10%.

Tabla 3.2 División del dataset.

DIVISIÓN ALEATORIA DEL DATASET	NÚMERO DE FOTOS
Entrenamiento (80%)	3682
Validación (10%)	460
Test (10%)	461

3.2 Prototipo del modelo y entrenamiento

Durante el entrenamiento se ejecutaron varias pruebas a diferentes valores de épocas, manteniendo constantes otros Hiperparámetros como el tamaño del Batch a 32, tamaño de la imagen a 640 píxel, tasa de aprendizaje en 0.01. En la tabla 3.3 se muestran varios resultados del tiempo de entrenamiento que tomó para cada prueba.

Tabla 3.3 Tiempo de entrenamiento.

NÚMERO DE ÉPOCAS	TIEMPO DE ENTRENAMIENTO
100 épocas	4.245 horas.
150 épocas	6.379 horas.
200 épocas	8.461 horas.
250 épocas	9.981 horas.

En capítulo anterior, en base a una publicación [22] se recomienda que un número adecuado de entrenamiento es 300 Épocas. Al ejecutar este proceso con el Dataset que se dispone para el proyecto, con ese número de épocas el modelo no finaliza el entrenamiento, ya que al iterar 250 épocas se detiene el entrenamiento por Early Stopping.

Para este caso vamos a analizar algunos resultados de métricas que se obtuvieron en este entrenamiento en las diferentes épocas, en la Figura 3.7 se muestran la función de Perdida para Bounding Box, Objetos y clases.

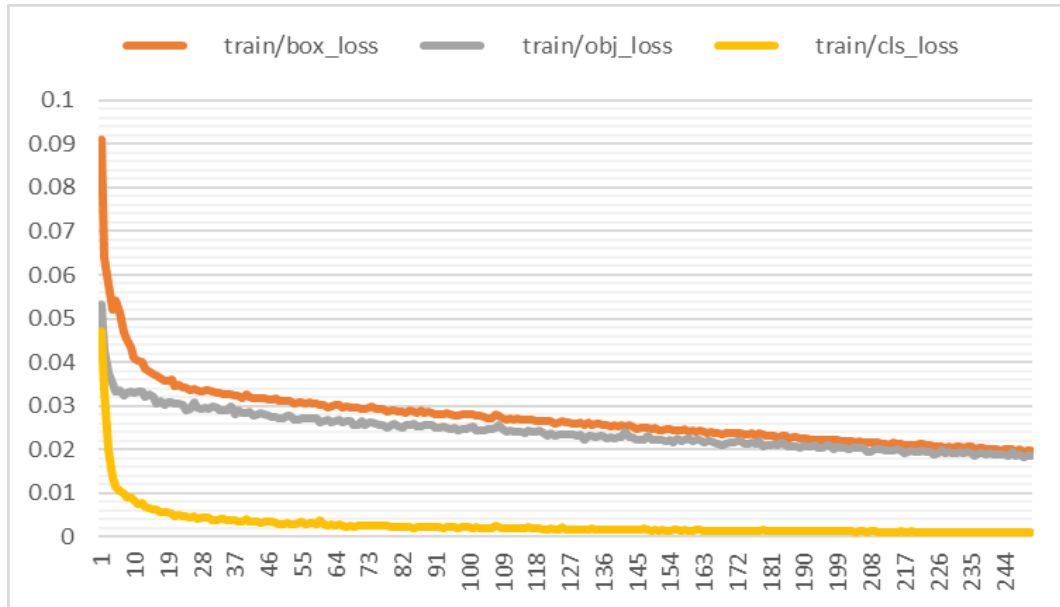


Figura 3.7 Función de pérdida.

Otro aspecto analizar en este proceso de entrenamiento son las métricas de Precisión y Recall. En la Figura 3.8 se muestra la evolución y diferencia entre estas dos en cada época.

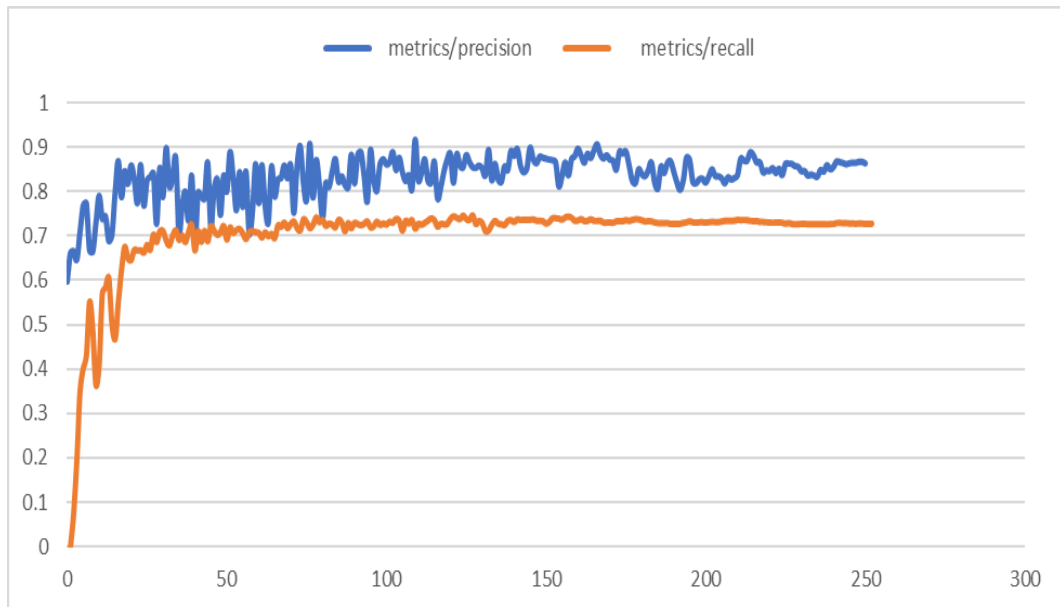


Figura 3.8 Precision Vs Recall.

En el entrenamiento se genera el mAP (mean Average Precision) esta métrica es importante por la relación de todas clases en función de los umbrales de intersección sobre Unión de las Áreas de Bounbding Box, para este caso en la Figura 3.9 se muestra su desarrollo durante cada época en cual fue entrenado el modelo.

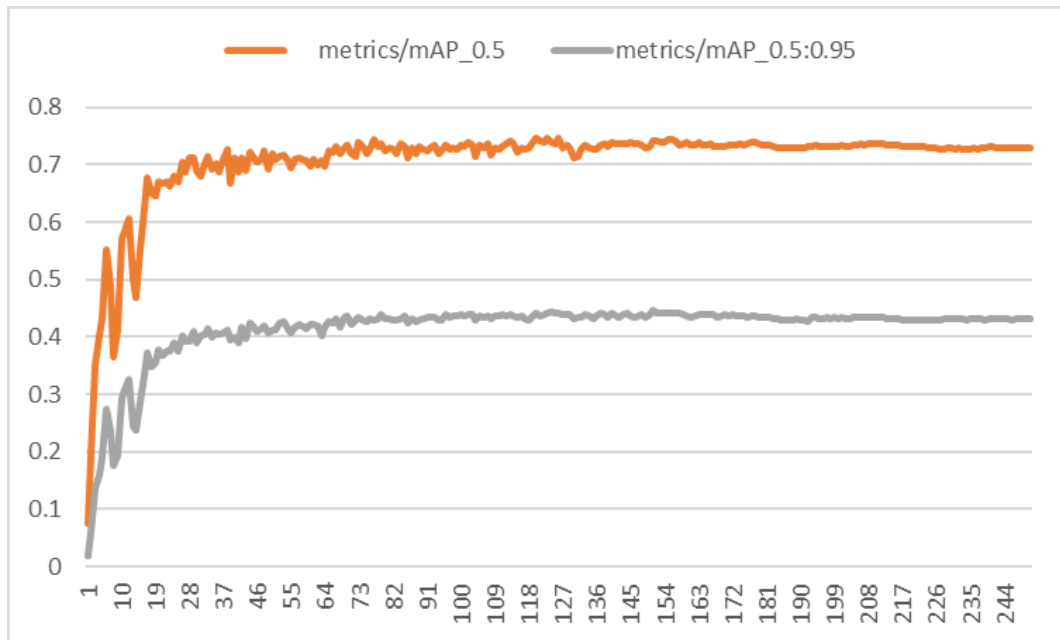


Figura 3.9 Métrica mAP (mean Average Precision).

Como se aprecia en las Figuras 3.9 el modelo tiene un comportamiento constante en la métrica a partir de la época 150 de entrenamiento, lo cual genera un evento de Early Stopping en el proceso, lo que quiere decir que se detiene el entrenamiento, ya que se ha superado el parámetro de Paciencia que en este caso esta seteado en un valor de 100 Epocas. Como detalle adicional este cuenta el número de épocas que se ha entrenado, sin tener una mejora en la métrica que se monitorea, en este caso es el mAP. Para el Dataset usado no es necesario continuar con el entrenamiento tratando de incrementar más épocas a este proceso.

3.3 Infraestructura para procesamiento y almacenamiento

La infraestructura consta de varios elementos que trabajan en conjunto para su funcionamiento, en el lado del nodo tenemos una cámara IP que se encarga de capturar el video del personal que ingresa a las instalaciones, esta información es enviada a otra ciudad donde se encuentra el data center de la empresa, a través de un enlace de datos L3 MPLS con una latencia de 13 milisegundos.

Una vez que la información llega al data center, debe pasar por un equipo de seguridad perimetral, el cual se encuentra configurado para permitir el tráfico de los protocolos necesarios. En la figura 3.10 podemos ver los puertos habilitados en la cámara.



Figura 3.10 Puertos configurados en la cámara.

Luego este tráfico es entregado al computador Nvidia DGX-A100 para que se procese el modelo de detección de cascos y mascarillas. Se ha utilizado la tecnología Docker para crear un contenedor que albergue el modelo, la base de datos y la interfaz gráfica. En la

figura 3.11 se muestra un diagrama de la infraestructura instalada para funcionamiento del proyecto.

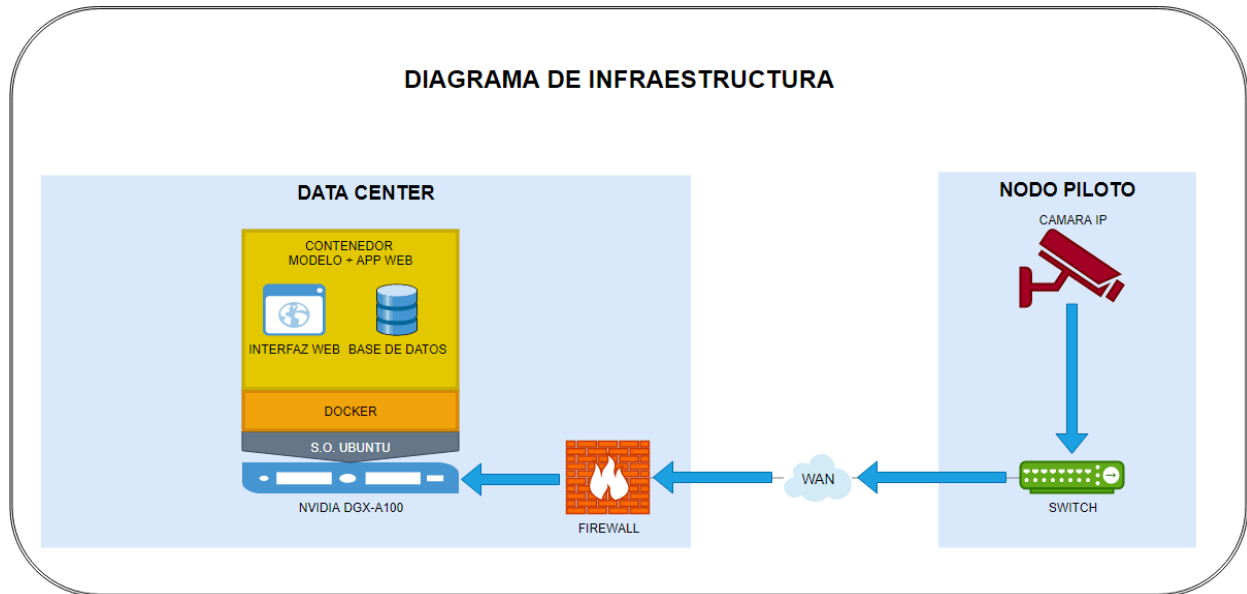


Figura 3.11 Diagrama de Infraestructura.

En la figura 3.12 podemos ver la instalación de la cámara en el nodo piloto, la cual realiza la captura de video de los técnicos que ingresan al nodo.



Figura 3.12 Instalación de cámara en nodo piloto.

En la tabla 3.4 se describen las características de la cámara instalada marca Huawei Modelo X2221-CL.

Tabla 3.4 Especificaciones Huawei Modelo X2221-CL.

Max resolution	2MP 1920(H)*1080(V) / 1080p
CODECs	MJPEG, H.265
Form Factor	Bullet
Sensor size	1/1.8" CMOS
Shutter speed	Supports slow shutter Slow shutter levels: 1/3, 1/4, 1/8, 1/12, 1/15 Non-slow shutter levels: 1/25, 1/30, 1/50, 1/60, 1/90, 1/100, 1/120, 1/125, 1/180, 1/250, 1/350, 1/500, 1/725, 1/1000, 1/1500, 1/2000, 1/3000, 1/4000, 1/6000, 1/10000, 1/30000, 1/100000, custom Default value: 1/25
Day/Night mode	Supported; automatic (ICR)/Color/B/W (default: color); Sensitivity level: high (default), medium, and low; Image triggering/Internal triggering
Gain control	Manual exposure: 101 levels configurable from 0 to 100 (step: 1; default: 0)
White balance	Auto/Manual/OnePush
Digital noise reduction	2D/3D noise reduction
WDR	120dB
Highlight compensation	Supported
Image stabilization	Supported Sensitivity levels are not supported.
Defogging	Supported
Exposure compensation	Supported
Exposure mode	Supported
SNR	>45dB
Motorized focus	Supported
Day/Night focus compensation	Supported
Backlight adaptation	Supported

Overcast adaptation	Supported
Speed adaptation	Supported
Corridor mode	Supported
Image correction	Vertical, horizontal, or both
Privacy mask	Supported
EMMC	Supported

En la figura 3.13 podemos ver los ajustes de Video configurados en la cámara Huawei Modelo X2221-CL.

Tipo de flujo

Modo espejo

Modo corredor

Velocidad de cuadros completa

Velocidad binaria dinámica (Kbps)

Velocidad de cuadros dinámica

Transmisión primaria Transmisión secundaria 1

Codificación inteligente

(Nota: La transmisión sin interrupciones y la región de interés no se aplican una vez habilitada la codificación inteligente.)

Protocolo de codificación

Complejidad de codificación

Resolución

Tipo de velocidad binaria

Velocidad binaria máxima (Kbps)

Velocidad de cuadros

Calidad de imagen

Intervalo de cuadro I

Codificación por capas

Marca de tiempo absoluta

Figura 3.13 Parámetros de video.

En la figura 3.14 podemos ver algunas configuraciones de red en la cámara, entre las más importantes la dirección IP, Mascara de Red, Gateway y MTU.

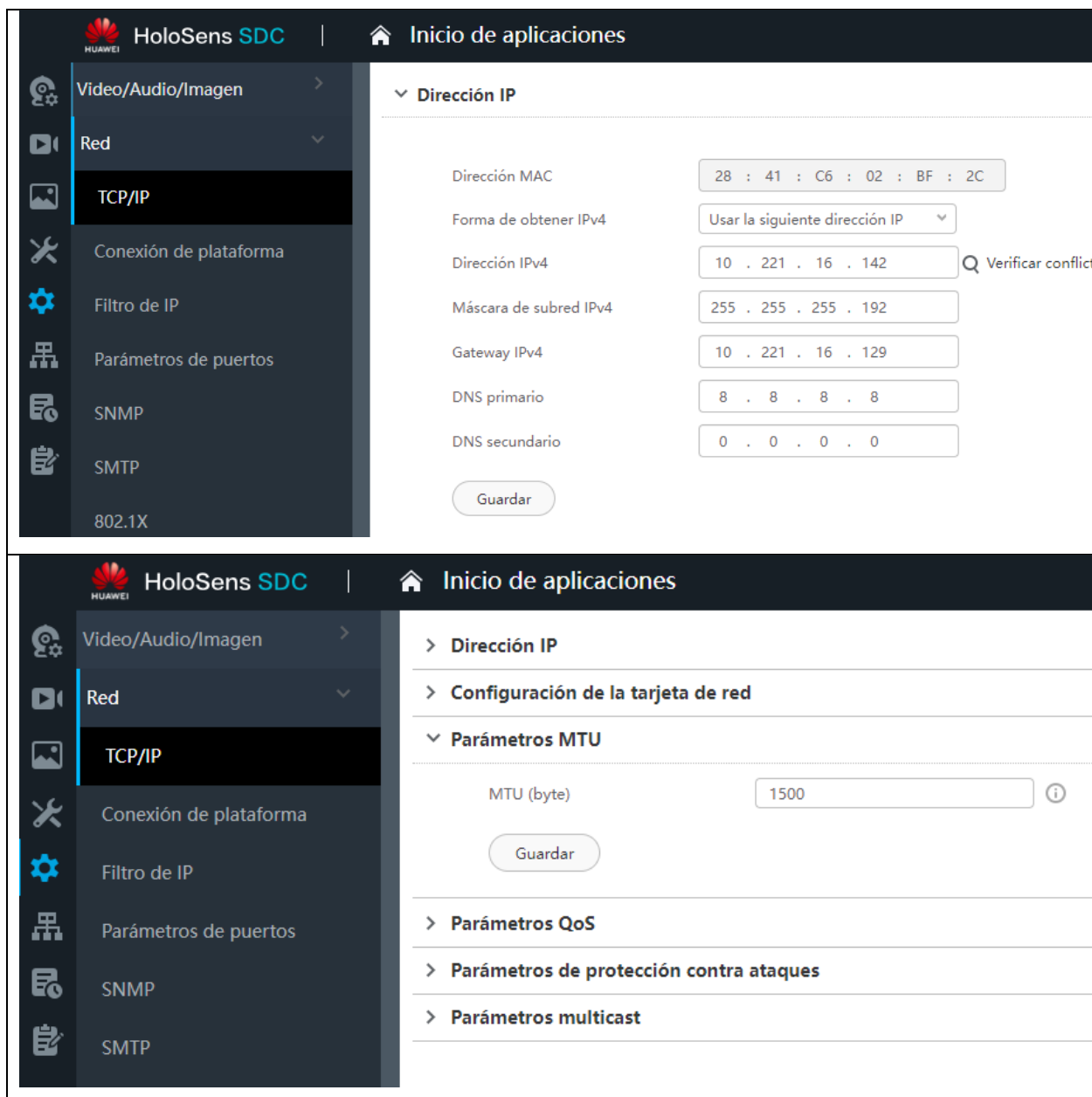


Figura 3.14 Parámetros de red.

En la figura 3.15 podemos ver la información del video en tiempo real obtenido con el software VLC Media Player 3.0.16. Se verifica que se está usando el codec H.265, una resolución de video de 1920x1080, y unos 29 cuadros por segundo aproximadamente.

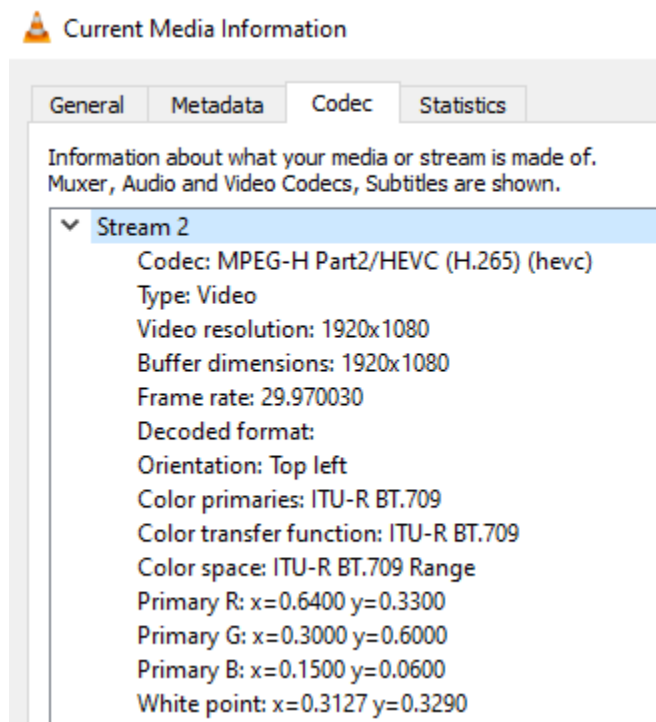


Figura 3.15 Información de la captura de video.

3.4 Interfaz gráfica y reportes

Se ha implementado una interfaz gráfica para la administración de las cámaras, esto nos va a ayudar a asociar las cámaras a los nodos, de tal manera que, al abrir una bitácora de acceso en los sistemas de la empresa, comience a ejecutar el modelo de detección.

Al ingresar al menú de CAMARAS, se mostrará la pantalla de la figura 3.16, la cual corresponde a la administración de cámaras, en este apartado podemos buscar una cámara.



Figura 3.16 Administración de cámaras.

Al dar clic en el botón NUEVA, se muestra la interfaz para ingresar una nueva cámara al sistema de monitoreo, ver figura 3.17, se ha descrito dentro de los cuadros de texto a que se refiere cada campo utilizando placeholders en html, cuando el usuario haga clic sobre cada cuadro de texto esta explicación desaparecerá automáticamente:

- **Nombre:** Nombre asignado a la cámara.
- **IP:** IP que se debe asignar a la cámara.
- **Puerto:** Puerto RTSP configurado en la cámara.
- **Usuario:** Nombre de usuario asignado a la cámara.
- **Password:** Clave del usuario asignado a la cámara.
- **Modelo:** Modelo de la cámara instalada en el nodo.
- **Serie:** Número de Serie de la cámara instalada en el nodo.
- **Path:** Complemento de la URL para acceder al video.
- **Ciudad:** Ciudad donde se encuentra instalada la cámara.
- **Nodo:** Nodo en el que se encuentra instalada la cámara.

CREAR NUEVA CAMARA

Nombre	IP
<input type="text" value="Nombre asignado a la camara"/>	<input type="text" value="Ej: 192.168.0.5/24"/>
Puerto	Usuario
<input type="text" value="Ej: 554"/>	<input type="text" value="Usuario de administracion de la camara"/>
Password	Modelo
<input type="text" value="Password de administracion de la camara"/>	<input type="text" value="Modelo de la camara"/>
Serie	
<input type="text" value="Nro. de serie"/>	
Path:	
<input type="text" value="Ej: /LiveMedia/ch1/Media1"/>	
Ciudad	Nodo
<input type="text" value="Quito"/>	<input type="text" value="NODOS5-QUITO"/>
<input type="button" value="GUARDAR"/>	

Figura 3.17 Formulario de creación de cámara.

La información guardada en el formulario anterior se utilizará para generar un link de conexión hacia la cámara, el formato se muestra en la tabla 3.5.

Tabla 3.5 Link de conexión a la cámara.

FORMATO
rtsp://usuario:password@IP:PUERTO/LiveMedia/ch1/Media1
EJEMPLO
rtsp://usuario_test:password_test@192.168.0.5:554/LiveMedia/ch1/Media1

En caso de necesitar modificar los datos de la cámara, se cuenta con un formulario para este propósito, para ello haremos clic sobre el botón MODIFICAR, como lo muestra la figura 3.18.

ADMINISTRACION DE CAMARAS

Ingresar Nombre Camara

Total de registros: 3

#	CIUDAD	NODO	NOMBRE	IP	MODELO	SERIE	ACCIONES
1	Quito	NODO1-QUITO	CAMARA1-HUAWEI	192.168.0.16/24	X2221-CL	2102411849LTK7000083	<input type="button" value="MODIFICAR"/> <input type="button" value="ELIMINAR"/>
5	Daule	NODOS-DAULE	CAMARA2-HUAWEI	192.168.0.17/24	X2221-CL	2102411849LTK7000084	<input type="button" value="MODIFICAR"/> <input type="button" value="ELIMINAR"/>

Previous Next

Figura 3.18 Seleccionar cámara a modificar.

Una vez hecho esto, se mostrará un formulario igual al de la figura 3.19, en donde podremos modificar los datos que necesitemos.

BITACORA

EDITAR CAMARA

Nombre IP

Puerto Usuario

Password Modelo

Serie

Path:

Ciudad Nodo

Figura 3.19 Modificar los datos de una cámara.

En el caso de que se tenga que retirar la cámara de un nodo, la interfaz gráfica cuenta con una opción de eliminar el activo, tal como lo muestra la figura 3.20.

BITACORA

ADMINISTRACION DE CAMARAS

Ingresar Nombre Camara

Total de registros: 3

#	CIUDAD	NODO	NOMBRE	IP	MODELO	SERIE	ACCIONES
1	Quito	NODO1-QUITO	CAMARA1-HUAWEI	192.168.0.16/24	X2221-CL	2102411849LTK7000083	<input type="button" value="MODIFICAR"/> <input type="button" value="ELIMINAR"/>
5	Daule	NODOS-DAULE	CAMARA2-HUAWEI	192.168.0.17/24	X2221-CL	2102411849LTK7000084	<input type="button" value="MODIFICAR"/> <input type="button" value="ELIMINAR"/>

Previous Next

Figura 3.20 Eliminar cámara.

En el caso de necesitar descargar un reporte, nos dirigimos al menú REPORTE, en donde se mostrará una interfaz como el de la figura 3.21. En este apartado deberemos buscar un nodo y la fecha en la que queremos visualizar el acceso, y presionamos el botón Buscar. Se listarán todos los accesos que se ha tenido en ese día y podremos descargar el reporte en formato pdf.

BITACORA

GENERACION DE REPORTES

CIUDAD	NODO	FECHA
Quito	NODO5-QUITO	Date: 2021-08-26
<input type="button" value="Buscar"/>		

#	CIUDAD	NODO	FECHA	HORA	CUADRILLA	ACCION
1	QUITO	NODO5-QUITO	2021-08-26	03:27 PM	CUADRILLA-UIO-00050	<input type="button" value="DESCARGAR REPORTE"/>

Figura 3.21 Generación de reportes.

En la figura 3.22 podemos ver el formato del reporte de acceso al nodo.

INFORME DE FISCALIZACION

FOR FIS 01 VER 1 (12 06 2019)

ANTECEDENTES:

N° INFORME: INF-20210826-0001

FECHA: 2021-08-26

HORA: 03:27 PM

CIUDAD: QUITO

NODO: NODO-UIO-G

DEPARTAMENTO RESPONSABLE: OPERACIONES URBANAS

CUADRILLA RESPONSABLE: CUADRILLA-UIO-00050

FISCALIZADOR ASIGNADO: PROYECTO IA

INFORME:

- Se realiza la fiscalización del ingreso al nodo, y se detecta el NO cumplimiento de las normas.

EVIDENCIA FOTOGRAFICA:

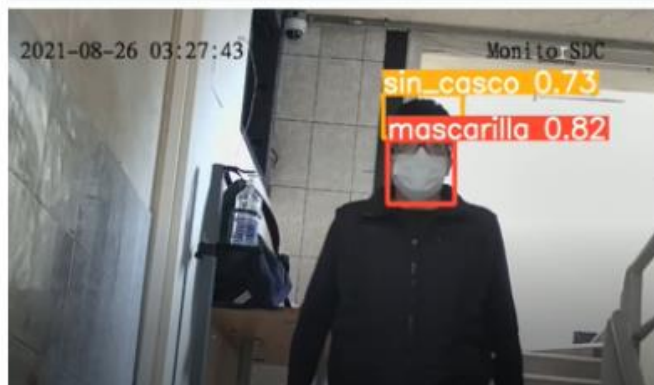


Figura 3.22 Reporte de acceso al nodo.

Adicionalmente, en la figura 3.23, podemos evidenciar el correo de alerta enviado al departamento de fiscalización, con el respectivo informe adjunto.

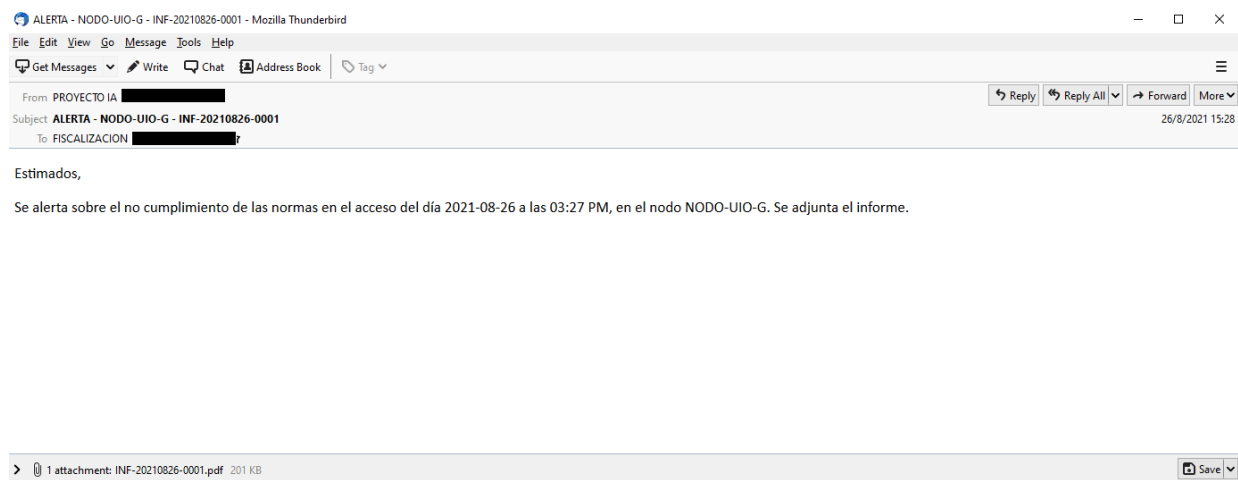


Figura 3.23 Envío de reporte vía email.

CAPÍTULO 4

4. ANÁLISIS DE RESULTADOS

4.1 Métricas y comunicación de resultados

Teniendo en cuenta que el personal de soporte en sitio son 1, 2 o 3 técnicos, más el agente de supervisión, genera una demora de 3 minutos adicionales en el ingreso según los informes internos de la empresa, este tiempo es tomado como un promedio de lo que el supervisor en oficina se demora en iniciar sesión en los sistemas de la empresa para ingresar a la cámara de supervisión. En la tabla 4.1 se expone un resultado sobre las horas hombre en minutos que se pierden en cada proceso de ingreso al nodo. Luego de la automatización este tiempo se reducirá a cero, debido a que el sistema de reconocimiento realizará este trabajo y generará el informe respectivo de manera automática.

Tabla 4.1 Revisión del departamento de fiscalización vs Sistema de reconocimiento.

NÚMERO DE TÉCNICOS	TIEMPO DE SUPERVISIÓN MANUAL (min)	HORAS HOMBRE PERDIDAS (min)	TIEMPO LUEGO DE AUTOMATIZACIÓN (min)
1	3	3	0
2	3	6	0
3	3	9	0

En la tabla 4.2 se presentan los resultados de los informes generados durante el mes de septiembre de 2021, de los cuales fueron objetados por el personal un total de 11 informes.

Tabla 4.2 Accesos al nodo en el mes de septiembre 2021.

NÚMERO DE ACCESOS	DETECCIONES CORRECTAS	DETECCIONES INCORRECTAS
113	102	11

4.2 Pruebas de funcionalidad

A continuación, se presentan algunos escenarios en los que se ha probado el sistema de detección, las imágenes muestran al personal ingresando al nodo en donde se ha implementado el primer piloto.

- **Persona ingresando con mascarilla y casco.**

En la Figura 4.1 podemos ver al técnico ingresando al nodo, con todos sus implementos, el algoritmo detecta el uso de mascarilla y casco con una probabilidad mayor al 70%.

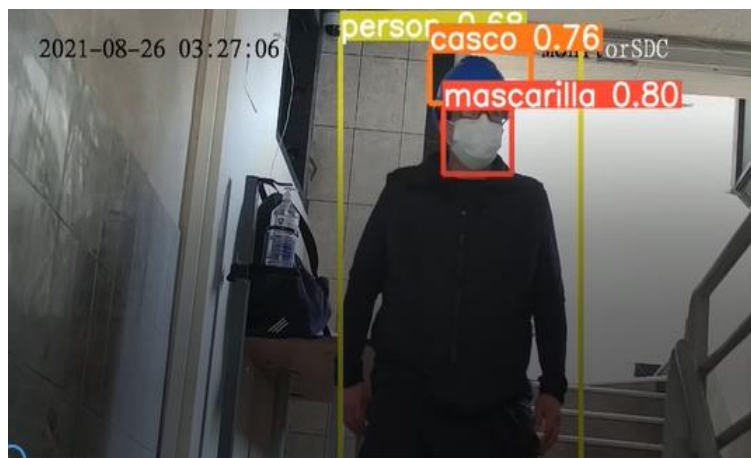


Figura 4.1 Técnico ingresando con mascarilla y casco.

- **Persona ingresando con mascarilla y sin casco.**

En la Figura 4.2 podemos apreciar al técnico ingresando sin casco y con mascarilla, con una probabilidad mayor al 70%.



Figura 4.2 Técnico ingresando con mascarilla y sin casco.

- **Persona ingresando con casco y sin mascarilla.**

En la Figura 4.3, observamos un incumplimiento de la norma en el uso de mascarilla, el algoritmo lo detecta con una probabilidad mayor al 80%.



Figura 4.3 Técnico ingresando con casco y sin mascarilla.

- **Persona ingresando sin casco y sin mascarilla.**

Uno de los escenarios que identifica el modelo que no cumple con las normas establecidas es cuando el técnico no cuenta con casco ni mascarilla. En la Figura

4.4 se muestra los valores de la probabilidad de 72% sin casco y un valor del 81% sin el uso de mascarilla.

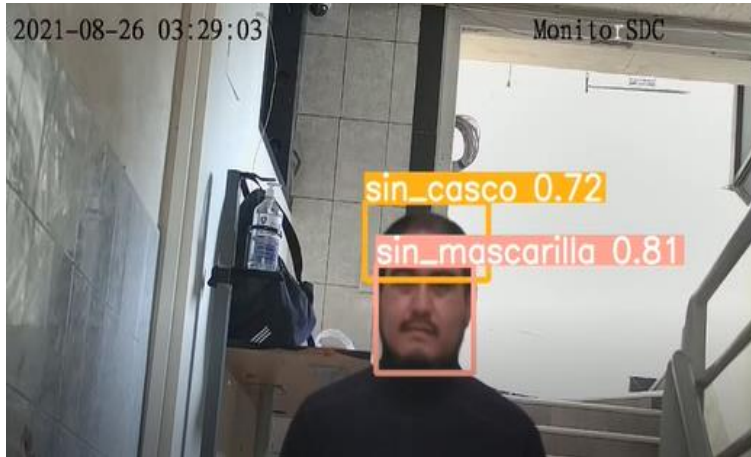


Figura 4.4 Técnico ingresando sin casco y sin mascarilla.

4.3 Análisis costo/beneficio

Actualmente el departamento de fiscalización cuenta con 40 personas que laboran en turnos rotativos para cubrir la demanda, debido a la expansión de la red el área ha seguido creciendo en recurso humano, lo cual lo hace insostenible a largo plazo.

En la tabla 4.3 vemos el gasto mensual para la operación del departamento de fiscalización, sin embargo, se debe tomar en cuenta que la persona de supervisión no utiliza todo su tiempo en esta actividad, por lo que posteriormente se realiza un análisis del tiempo de supervisión por parte del personal de fiscalización.

Tabla 4.3 Costo mensual en recurso humano.

PERSONAS	SUELDO (dólares)	COSTO MENSUAL (dólares)
40	800	32.000

Según un informe interno de la empresa, cada persona del departamento de fiscalización dedica el 33% de su tiempo en este proceso. En la tabla 4.4 se detalla el gasto mensual que se podría reducir por la implementación del proyecto.

Tabla 4.4 Costo mensual del proceso de supervisión.

PERSONAS	COSTO DE SUPERVISIÓN (33% del sueldo en dólares)	COSTO MENSUAL (dólares)
40	264	10.560

La tabla 4.5 describe la inversión inicial que se debe realizar para que cada nodo cuente con el hardware necesario que permita obtener el video.

Tabla 4.5 Inversión inicial.

CONCEPTO	CANTIDAD	COSTO UNITARIO (dólares)	INVERSIÓN INICIAL (dólares)
Cámaras	350	200	70.000
Instalación	350	10	3.500
		TOTAL	73.500

Si quisiéramos recuperar la inversión en un año, tenemos que el costo mensual de la inversión inicial es \$73.500 dividido entre 12 meses, lo cual nos da un valor de \$6.125 mensuales, comparado con el costo de \$10.560 que cuesta hacerlo de manera manual, se estima un ahorro de \$4.435 mensuales. Para el segundo año ya no es necesaria esta inversión, por lo que se observa un importante ahorro para la empresa.

Como la empresa cuenta con la infraestructura a nivel de comunicaciones, Datacenter y equipos de cómputo como DGX-A100 el costo de la implementación se reduciría, ya que todos los componentes ya se encuentran implementados a excepción de las cámaras. Por lo cual el despliegue hacia los demás nodos se lograría de una manera rápida disminuyendo tiempo y costo sobre este proyecto.

Debido al giro de negocio de la empresa, los minutos adicionales ganados representan grandes beneficios a los clientes finales, como ejemplo podemos poner a empresas que participan en licitaciones, si la propuesta de una licitación se debía subir hasta las 23:00, pero el servicio se recuperó 23:02, esos 2 minutos son una diferencia grande, en donde se podría perder negocios importantes. Si

bien es cierto el beneficio en términos monetarios si es tangible, debido a su naturaleza es difícil estimarlo y no se lo realizará en este proyecto.

5. CONCLUSIONES Y RECOMENDACIONES

- Se ha observado que la mayoría de los informes se generaron de manera adecuada, sin embargo, estos podrían ser mejorados si incrementamos el dataset de entrenamiento, es decir, tener imágenes más variadas en cuanto al ángulo de la cámara que toma las fotos de los técnicos y con distinta iluminación del lugar. Adicionalmente podemos utilizar técnicas de data augmentation para este fin.
- Según los datos obtenidos por la herramienta de acceso a los nodos, se ha conseguido una precisión de 90,2 % de informes bien generados.
- Para un mejor rendimiento de la aplicación las cámaras que se instalen en los nodos deberán ser ubicadas a una altura adecuada, de tal manera que capturen el ingreso de los técnicos casi de frente, esto mejorará el funcionamiento del sistema ya que fue entrenado bajo estas condiciones.
- Se recomienda usar un servidor NTP para sincronizar la fecha y hora de todas las cámaras del proyecto y el servidor donde se genera los logs.
- El trabajo realizado de diseño y montaje de infraestructura, pruebas del modelo en tiempo real, es la columna vertebral sobre la cual se desarrollarán nuevos proyectos dentro de la empresa, y también será integrado como parte primordial en la automatización de todo el proceso de acceso a las instalaciones, de tal manera que se pueda reducir el personal en oficinas que interactúa con el técnico en el nodo.
- El módulo de generación de reportes puede contribuir a otros departamentos de la empresa que necesiten información acerca de fecha hora que fue realizada una instalación o soporte y realizar una evaluación continua de cumplimiento de la planificación diaria con el personal. Toda esta información puede ser accedida no solo en forma manual sino a través de interacciones con algunos sistemas que dispone la compañía lo que potencia aún más el proyecto.

REFERENCIAS

- [1] Who.int. [Online]. Available: <https://www.who.int/es/emergencies/diseases/novel-coronavirus-2019/advice-for-public/when-and-how-to-use-masks>. [Accessed: 13-Sep-2021].
- [2] Kaggle.com. [Online]. Available: <https://www.kaggle.com/andrewmvd/face-mask-detection/download>. [Accessed: 10-Sep-2021].
- [3] “Kaggle: Your Home for Data Science,” Kaggle.com. [Online]. Available: <https://www.kaggle.com/andrewmvd/hard-hat-detection/download>. [Accessed: 10-Sep-2021].
- [4] “SpeedFace-V5L [TD],” Zkteco.eu, 25-Mar-2020. [Online]. Available: <https://zkteco.eu/products/body-temperature-measurement/speedface-v5l-td>. [Accessed: 05-Sep-2021].
- [5] I. P. L. Pequeños, “Solución de Detección de Mascarilla Solución de Detección de Mascarilla,” Hikvision.com. [Online]. Available: <https://www.hikvision.com/content/dam/hikvision/es-la/brochures-esp/Infograf%C3%ADa-Soluci%C3%B3n-de-detecci%C3%B3n-de-mascarilla.pdf>. [Accessed: 05-Sep-2021].
- [6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” arXiv [cs.CV], 2015.
- [7] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” arXiv [cs.CV], 2020.
- [8] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
- [9] “NVIDIA DGX A100,” Nvidia.com. [Online]. Available: <https://www.nvidia.com/en-us/data-center/dgx-a100/?ncid=pa-srch-goog->

877823&gclid=CjwKCAjwhOyJBhA4EiwAEcJdcVwzklcsebrPDiQSAb_frzeoStrfO6Rw6PkxR6oDBO5hu0QkCTNCnxoCgtEQAvD_BwE. [Accessed: 10-Sep-2021].

[10] “CUDA Zone,” Nvidia.com, 18-Jul-2017. [Online]. Available: <https://developer.nvidia.com/cuda-zone>. [Accessed: 14-Sep-2021].

[11] “Annotation tool,” Superannotate.com, 14-Apr-2021. [Online]. Available: <https://superannotate.com/annotation-tool/>. [Accessed: 05-Sep-2021].

[12] “The best image labeling tools for Computer Vision,” Dida.do, 27-Apr-2021. [Online]. Available: <https://dida.do/blog/the-best-labeling-tools-for-computer-vision>. [Accessed: 14-Sep-2021].

[13] M. Massiris, C. Delrieux, and J. Á. Fernández, “Detección de equipos de protección personal mediante red neuronal convolucional YOLO,” in Actas de las XXXIX Jornadas de Automática, Badajoz, 5-7 de septiembre de 2018, 2020.


[14] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth, “SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2,” Sustain. Cities Soc, vol. 66, no. 102692, p. 102692, 2021.

[15] J. leamsaard, S. N. Charoensook, and S. Yammen, “Deep learning-based face mask detection using YoloV5,” in 2021 9th International Electrical Engineering Congress (iEECON), 2021.

[16] “Empowering app development for developers,” Docker.com. [Online]. Available: https://www.docker.com/?utm_source=google&utm_medium=cpc&utm_campaign=dockerhomepage&utm_content=namer&utm_term=dockerhomepage&utm_budget=growth&gclid=Cj0KCQjw4eaJBhDMARIsANhrQADUAB9bKXAfaYXUs-6ffJez3x3Uz72POxrvBiXRIB_w-RPqE2phuScaAg3SEALw_wcB. [Accessed: 10-Sep-2021].

[17] A. Soloaga and V. T. las P. de Ana Soloaga, “Python, los 5 usos más importantes de este lenguaje de programación,” Akademus.es, 19-Oct-2018. [Online]. Available:

<https://www.akademus.es/blog/programacion/principales-usos-python/>. [Accessed: 10-Sep-2021].

[18] darrenl, labellmg:  Labellmg is a graphical image annotation tool and label object bounding boxes in images.

[19] “Qué es Flask,” Openwebinars.net, 17-Nov-2017. [Online]. Available: <https://openwebinars.net/blog/que-es-flask/>. [Accessed: 10-Sep-2021].

[20] “MariaDB,” Capterra.ec. [Online]. Available: <https://www.capterra.ec/software/132005/mariadb>. [Accessed: 10-Sep-2021].

[21] Tzutalin. Labellmg. Git code (2015). <https://github.com/tzutalin/labellmg>

[22] J. leamsaard, S. N. Charoensook, and S. Yammen, “Deep learning-based face mask detection using YoloV5,” in 2021 9th International Electrical Engineering Congress (iEECON), 2021.