UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

MONITOREO DE SEGURIDAD Y MEDIDAS DE PREVENCIÓN DE CONTAGIO DE COVID-19 MEDIANTE DETECCIÓN EN SISTEMA CCTV PARA LA MINERÍA

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

JOSÉ IGNACIO MUSSO ZAPICO

PROFESOR GUÍA:
JAVIER RUIZ DEL SOLAR SAN MARTÍN

MIEMBROS DE LA COMISIÓN:
CESAR AUGUSTO AZURDIA MEZA
MAURICIO CORREA PEREZ

SANTIAGO DE CHILE
2021

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE MAGÍSTER EN CIENCIAS
DE LA INGENIERÍA
POR: **JOSÉ IGNACIO MUSSO ZAPICO**
FECHA: 2021
PROF. GUÍA: JAVIER RUIZ DEL SOLAR SAN MARTIN

MONITOREO DE SEGURIDAD Y MEDIDAS DE PREVENCIÓN DE CONTAGIO DE COVID-19 MEDIANTE DETECCIÓN EN SISTEMA CCTV PARA LA MINERÍA

La pandemia generada por el virus COVID-19 ha afectado enormemente a la industria minera: 275 minas debieron cesar sus operaciones en algún momento a lo largo del globo, conllevando a pérdidas millonarias. Es por esto que, soluciones orientadas a la prevención de contagios son cada ve más requeridas de parte de las empresas mineras.

En este trabajo se desarrollan cuatro módulos de visión computacional diseñados específicamente para monitorear y, a la vez, reducir los riesgos operacionales y los de contagio de COVID-19. Abordando los aspectos críticos de: distanciamiento social, uso de mascarillas, uso de casco y acceso a zonas riesgosas restringidas.

Para lograr un desempeño competitivo, en relación al gran volumen de investigaciones existentes en esta área, el trabajo, desarrollado en base a redes neuronales, se sitúa en el estado del arte de la disciplina de *machine learning*, utilizando las redes neuronales *CenterNet* y *YOLOv5* junto con técnicas de *tracking* como *deepSORT*. Esto permitió generar detectores con alta precisión, junto con una gran velocidad de inferencia en cada uno de los cuatro desafíos planteados.

El módulo detector de distanciamiento social construido es capaz de detectar el 72 % de los incumplimientos en las escenas de test. El módulo de zonas restringidas es capaz de identificar hasta el 92 % de las personas en las escenas, posicionándolo como un sistema competitivo con el estado del arte.

Para los módulos detectores de uso de mascarillas y elementos de protección personal, se generaron bases de datos nuevas, correspondientes a ensambles de datos recopilados, alcanzando más de 5000 anotaciones para el modelo de mascarillas. En cuanto al desempeño, el detector de uso de mascarillas alcanza una precisión de 82.3 % y *recall* de 74 % en ciertos puntos de operación, cifras comparables con estudios contemporáneos en Hong Kong. Además, el modelo detector de elementos de protección personal obtuvo un mAP de 80 % superando varios detectores actuales.

Se consigue así una plataforma de apoyo a la seguridad y prevención de COVID-19, potencialmente aplicable a cualquier faena o mina, automatizando un proceso preventivo crucial en términos humanos y económicos.

Ante todo, mucha calma.

Agradecimientos

Gracias a mis padres, Claudia y Roberto, a quienes admiro enormemente, por su constante motivación, apoyo incondicional y siempre estar ahí.

Gracias a todos mis hermanos por crecer juntos. Gracias a mis abuelos por transmitirme su orgullo.

Gracias a todos mis amigos quienes siempre me aportaron energía y vitalidad.

Gracias a los integrantes de mi comisión por guiarme en el transcurso la investigación y compartir conmigo su conocimiento. Gracias a PSINet por los recursos otorgados y al AMTC por su apoyo.

Tabla de Contenido

Índice de Ilustraciones

Capítulo 1

Introducción

A principios del 2020 era evidente que se acercaba una época de disrupción en la industria minera, trayendo consigo grandes cambios, principalmente en el ámbito digital y tecnológico. Sin embargo, nadie imaginaba que esta transformación llegaría de la mano de una pandemia global como lo ocurrido con el virus COVID-19.

Los cambios que trajo esta crisis sanitaria global llegaron para quedarse no sólo en el sector de la minería, sino en todos los sectores. Así ocurre con los cambios en el sector sanitario, respecto a los desafíos de capacidad; el sector educacional, respecto al aprendizaje remoto; y el sector del turismo, en relación al traslado de las atracciones al mundo virtual.

Según estudios realizados por la consultora EY[1], los riesgos y oportunidades que predominarán este 2021 involucran conceptos esenciales para la supervivencia a la pandemia del Coronavirus. Como se observa en a la Figura 1.1, los top 3 reisgos u oportunidades son: Licencia para Operar, riesgos de alto impacto y productividad y aumento de costos. Estos tres elementos denotan que la transformación digital dejó de ser una alternativa y pasó a tomar un rol esencial en la subsistencia de las compañías mineras. Ya sea para supervisar normas, para prever potenciales riesgos o para optimizar procesos y reducir riesgos, los avances tecnológicos están internándose en esta industria a pasos agigantados.

[1] https://www.ey.com/en_gl/mining-metals/top-10-business-risks-and-opportunities-for-mining-and-metals-in-2021

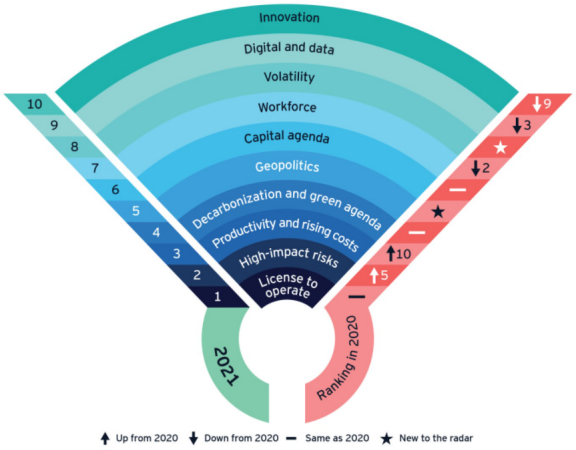


Figura 1.1: Top 10 riesgos y oportunidades en la industria minera el 2021

El objetivo principal de este proyecto consiste en desarrollar un sistema de monitoreo a través de inteligencia artificial para proveer seguridad operacional y prevenir contagios del virus COVID-19. Distanciamiento social, uso de mascarilla, de casco y vigilancia de zonas restringidas son los cuatro conceptos que serán monitoreados por los módulos a desarrollar en esta memoria. Se ensamblaron estos cuatro módulos mediante la implementación de modelos del estado del arte de visión computacional.

Estos cuatro sistemas atacan directamente los principales riesgos de sector metalúrgico al disminuir riesgos laborales y sanitarios. El rendimiento de cada módulo fue cuantificado con el objetivo de determinar el beneficio de su implementación obteniendo resultados positivos. Esto podría convertir este trabajo en un potencial avance hacia una nueva "Minería Inteligente".

Motivación

Identificación y formulación del problema

El 2020 estudios[1] indicaron que, los principales riesgos que aquejaron a la industria minera corresponden a la Licencia para Operar y la Efectividad Digital. El primer concepto es una licencia que define aspectos ambientales, sociales y de resultados, requeridos para operar; mientras el segundo, tiene relación con la aplicación de soluciones digitales en la cadena de valor dentro de la minería. La presencia de éstos dos conceptos en lo más alto de los riesgos, da cuenta que ésta industria enfrenta actualmente una severa disrupción donde el foco se vio desplazado hacia la colaboración con la sociedad y la transformación digital. Será clave, entonces, que las mineras se adapten digitalmente en los próximos años, para sobrevivir a la competencia.

De acuerdo con la circular N° 3336[2] de la Superintendencia de Seguridad Social, en caso de accidente grave o fatal, la empresa debe suspender las faenas afectadas y, de ser requerido, evacuar el lugar de trabajo. Además de las fatales pérdidas humanas, esto conlleva pérdidas

de millones de dólares. Por consiguiente, el uso de sistemas enfocados en el cumplimiento de normas para evitar accidentes trae beneficios tanto en el ámbito regulatorio como en el preventivo.

Otro aspecto preocupante para la industria minera hoy en día corresponde a la pandemia generada por el virus COVID-19. Este fenómeno ya ha afectado la industria al repercutir en el precio del cobre, produciendo una baja de más del 19 % en los meses de mayo y junio del 2020, como fue estipulado por Manuel Viera, Presidente de la Cámara Minera de Chile, en una entrevista para Revista Energía[3]. Sin embargo, la toma de medidas de seguridad respecto al virus, ha permitido que las faenas no cesen sus operaciones. Esto demuestra el interés de las compañías en la prevención de contagios dentro de la industria, ya que los efectos pueden ser devastadores.

Al combinar las amenazas descritas a la industria minera con la actual disrupción digital, surge un producto que llega para cambiar los paradigmas estructurales: inteligencia artificial aplicada al monitoreo y seguridad en las minas.

El trabajo a realizar consiste en implementar un sistema de seguimiento de seguridad y prevención de contagios de COVID-19, para ser montado sobre el CCTV de una faena de CODELCO, proporcionado por la empresa PSINET.

Objetivos

Objetivo General

Desarrollar un sistema de visión computacional para el monitoreo de medidas preventivas de contagio de COVID-19 y reducción de riesgos.

- Seguridad:
  - Prevención de riesgos
  - Control de zonas restringidas y de peligro
- COVID-19:
  - Prevención de contagios
  - Cumplimiento de sugerencias sanitarias

Para alcanzar los objetivos se desarrolla un sistema con múltiples módulos utilizando visión computacional mediante las redes neuronales *CenterNet*[4] y *YOLOv5[5]* al tener un desempeño comparable con el estado del arte en tareas de de detección de objetos.

El sistema es probado con videos del CCTV de la empresa CODELCO, proporcionados por PSINET.

Objetivos específicos

- Obtención de base de datos de cámaras dentro de las mineras
- Entrenamiento de red *YOLOv5*[5] para la detección mascarillas y cascos.
- Uso de la red *CenterNet*[4] para identificar peatones.
- Detección exitosa de peatones, elementos de protección personal (EPP) y mascarillas en escenas.
- Generación de la heurística necesaria para las cuatro aplicaciones.
- Recopilar datos de interés como rostros cuando se infringe alguna de las normas.
- Obtención de videos demostrativos.

Estructura del Documento

Explicados los objetivos principales y la motivación tras estos, se procede a contextualizar, en los capítulos 2 y 3, mediante una descripción del problema a abordar y un marco teórico junto con una descripción del estado del arte de la detección de objetos. Posteriormente se detallan cada uno de los cuatro módulos, en los capítulos 4, 5 y 6, donde se presentan: metodología, resultados y análisis de resultados. Para finalizar, en los capítulos 7 y 8, con una conclusión que recapitula el sistema implementado y se plantea un potencial trabajo a futuro.

Capítulo 2

Problema a abordar

2.1. Antecedentes

En Chile, la industria minera es la que posee en promedio la mayor cantidad de días perdidos al año por cada accidente (38.3 días en 2019)(Ver Figura 2.1) [6]. Cada uno de estos días conlleva pérdidas millonarias, según la producción de cobre de mina por empresa [7]. Es por esto que la seguridad toma un rol importantísimo en esta industria buscando resguardar la integridad física de los trabajadores.

Promedio de días perdidos por cada accidente del trabajo según actividad económica
Mutualidades
2010 - 2019

Actividades Económicas	2010	2019	Var %
Minería	31,4	38,3	22%
Transporte y Comunicaciones	16,8	25,5	52%
Construcción	17,7	23,0	30%
EGA[1]	16,1	22,3	39%
Industrias Manufactureras	15,7	21,7	38%
Agricultura y Pesca	14,4	21,5	49%
Promedio Nacional	14,2	19,7	39%
Servicios	12,0	17,3	45%
Comercio	11,6	16,0	38%

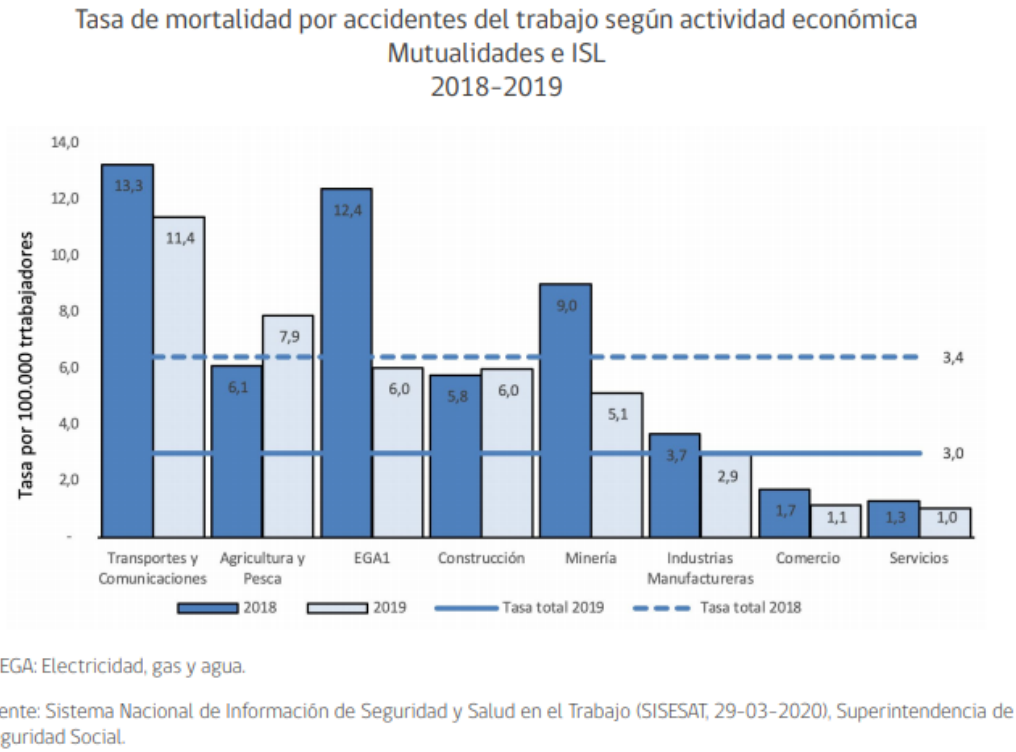
[1] EGA: Electricidad, gas y agua.

Fuente: Boletines Estadístico, Superintendencia de Seguridad Social.

Figura 2.1: Promedio de días perdidos por cada accidente del trabajo según actividad económica [6]

Si bien existen varias normas y reglamentos que rigen los estándares a seguir para efectuar una operación segura, es imposible que estos abarquen cada uno de los aspectos de riesgo. Con el fin de corroborar que se cumplen estas normas, se efectúan fiscalizaciones periódicas

para verificar el cumplimiento de estas normas. Además de estas fiscalizaciones, en la actualidad existen múltiples avances en el ámbito de la seguridad y monitoreo dentro de las minas. Sin embargo, esta industria aún se encuentra sobre el promedio nacional de tasa de mortalidad por accidentes, tal como se aprecia en la Figura 2.2[6].



[1] EGA: Electricidad, gas y agua.

Fuente: Sistema Nacional de Información de Seguridad y Salud en el Trabajo (SISESAT, 29-03-2020), Superintendencia de Seguridad Social.

Figura 2.2: Tasa de mortalidad por accidentes del trabajo según actividad económica [6]

2.2. Zonas Restringidas y uso de Elementos de Protección Personal

Como se detalla en la "Campaña de Seguridad Minera" realizada por el SERNAGEOMIN, el 75 % de las muertes ocurridas desde el 2000 al 2016 se adjudican a los siguientes tipos de accidentes:

- Golpeado por roca
- Caída desde altura
- Accidentes ocasionados por vehículos motorizados
- Atrapamiento

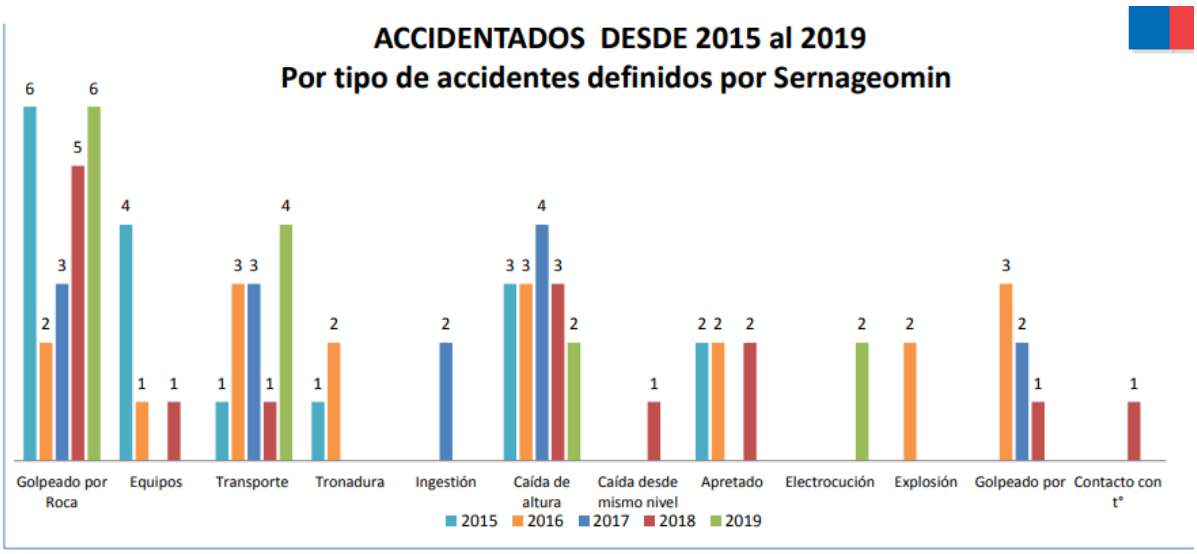
(a) (b)

(c) (d)

Figura 2.3: Campaña de SERNAGEOMIN para concientizar sobre las principales causas de accidentes

Respecto a los accidentes mencionados, se observa que las reglas de prevención de riesgos, tienen dos factores que se repiten constantemente: Uso de Elementos de Protección Personal (EPP) y el control de acceso a áeras de peligro o restringidas.

Por un lado, el uso de EPP queda regido por la "Normativa de Seguridad Minera" dictada por el SERNAGEOMIN [8], donde en múltiples artículos se detalla el uso de estos. Sin embargo, no siempre se siguen las normas al pie de la letra, lo que genera un aumento en el potencial riesgo al que se exponen los trabajadores.



Golpeado por roca: Accidentes cuya causa inmediata sea la caída de roca, incluyendo los derrumbes y colapso por subsidencias.
Equipo: Accidentes cuya causa inmediata esta asociada a la participación de un equipo minero.
Transporte: Accidentes cuya causa inmediata esta asociada a la participación de un equipo de transporte, como transporte de mineral, transporte de personal, transporte de agua camión aljibe, etc.
Ingestión : Accidente cuya causa inmediata sea la ingestión de líquidos, toxico y no toxico.
Tronadura : Accidentes cuya causa inmediata sea la proyección de partícula u onda expansiva producto a una tronadura, incluyendo los tiros quedados.
Caída de altura: Accidentes cuya causa inmediata corresponde a la caída de diferente nivel de altura.
Caída de mismo nivel: Accidentes cuya causa inmediata corresponde a la caída del mismo nivel de altura.
Falta de oxigeno: Accidentes cuya causa inmediata sea la baja concentración de oxigeno o la presencia de gases tóxicos.
Apretado por: Accidentes cuya causa inmediata sea el aprisionamiento del cuerpo o parte de el.
Electrocución: Accidentes cuya causa inmediata sea el contacto con energía eléctrica.
Explosión: Accidentes cuya causa inmediata sea la explosión de aparato a presión, acumulaciones de gases o explosiones generadas por explosivos excluyendo la tronadura, Ejemplo Explosión en un polvorín.
Golpeado por: Accidentes cuya causa inmediata sea producto al golpe causado por el impacto de algo material a una persona.
Contacto a T° : Accidentes cuya causa inmediata sea el contacto con temperatura.

Figura 2.4: Accidentados por tipo desde el 2015 al 2019 [9]

Por otro lado, si se observa la Figura 2.4, tanto el tipo "Golpeado por roca" como "Caída de altura" se asocian con el hecho que el afectado se encuentre en un área de alta peligrosidad o riesgo, entre otros factores. Para mitigar esto, se emplean distintas medidas y barreras físicas para generar distancia entre el trabajador y la zona de peligro, pero aun así se generan accidentes al no existir un sistema de alarma en tiempo real.

2.3. Medidas sanitarias para evitar contagios por COVID-19

Al 6 de julio de 2020, CODELCO, la primera productora de cobre del mundo, presenta más de 2.600 operarios contagiados por COVID-19, de los cuales más de 400 corresponden solo a la mina Chuquicamata[10]. A esto se le suma las 9 personas ya fallecidas, por el mismo virus, pertenecientes al sector minero. Esto indica que la industria minera ya se está viendo afectada por la pandemia global. Chuquicamata hoy opera a un tercio de su capacidad para prevenir el contagio dentro del personal. Sin embargo, aún existe un gran descontento dentro de los empleados respecto a la carencia de medidas de sanidad de parte de la empresa.

Figura 2.5: Distanciamiento social y uso de mascarillas en minas

Con el objetivo de combatir esta pandemia es que tanto el Servicio Nacional de Geología y Minería, como las empresas socias del Consejo Minero, han tomado medidas sanitarias para evitar contagios. A continuación se enumeran algunas.

- Uso obligatorio de mascarilla o escudo facial
- Distanciamiento social de al menos 1 metro
- Mejora en la calidad de higiene del lugar de trabajo
- Turnos flexibles de 14x14
- Controles preventivos de salud frecuentes
- Suspensión de reuniones masivas

La fiscalización de estas medidas se traduce en que al 21 de junio de 2020 se han realizado más de 1.400 fiscalizaciones a más de 4.600 instalaciones de faenas mineras[11].

(a) (b)

Figura 2.6: Campaña del Consejo Minero para la prevención de contagios

2.4. CODELCO y Circuito Cerrado de Televisión provisto por PSINET

Como se mencionó anteriormente, el sistema se implementará sobre el circuito cerrado de televisión (CCTV) de una mina de la empresa CODELCO. CODELCO es la principal empresa y motor del desarrollo de Chile. Es una empresa estatal y posee el mayor nivel de reservas y recursos de cobre del mundo[12]. Tal envergadura a nivel operacional implica una gran complejidad para cada uno de los procesos, es por esto que su CCTV es administrado por la empresa PSINet, la cual facilitó la base de datos para el desarrollo del proyecto.

PSINet es una corporación que ofrece servicios de integración en Infraestructura Tecnológica, Servicios de TI, Monitoreo, Control y Analítica. Entre sus proyectos se encuentra el servicio de suministro e instalación CCTV de CODELCO.

Los videos capturados por las cámaras ubicadas en las minas son almacenados por PSINet, por lo que existe una base de datos con videos históricos de más de 600 cámaras de vigilancia. En el proyecto se trabaja con parte de estos datos, en particular, con videos de tres zonas distintas para los detectores de distanciamiento y de mascarilla.

Capítulo 3

Marco Teórico y Estado del Arte

3.1. Detección de objetos: Historia y detalle de las redes CenterNet y YOLOv5

La detección de objetos en escenas, ya sea en imagen o video, corresponde a identificar el tipo de objeto y su posición dentro de la imagen. Es posible dividir la historia de la detección de objetos en dos periodos: "periodo de detección de objetos tradicional" y "periodo de detección de objetos basada en *deep learning*"[13]. En el primer periodo surgen técnicas basadas en la detección de características, las cuales, mediante el uso de detección de patrones locales, ventanas deslizantes, filtros y gradientes son capaces de identificar objetos en escenas en distintas escalas y orientaciones. Empleando estas técnicas se obtuvieron algoritmos como los detectores de patrones binarios locales (LBP)[14], SIFT[15], histogramas de gradientes orientados (HOG)[16] y modelo de partes deformables (DPM)[17].

Estos modelos significaron grandes avances en el ámbito de la detección de objetos, además de inspirar modelos siguientes. Sin embargo, estos se limitan a la detección de características *hand-crafted* o *hechas a mano* por lo que su capacidad de generalización depende de la habilidad de las personas de traducir características reales a patrones representables. Es así, como entre 2010 y 2014 se lleva a cabo la transición hacia el "periodo de detección de objetos basada en *deep learning*". Gracias a una mayor disponibilidad de poder de cómputo y avances en torno a la investigación de redes neuronales, se genera un "renacer" de las redes neuronales convolucionales (CNN). Con la propuesta de la red RCNN[18], R. Girshick *et al.* abre el paso hacia el desarrollo de la detección de objetos basada en *deep learning*.

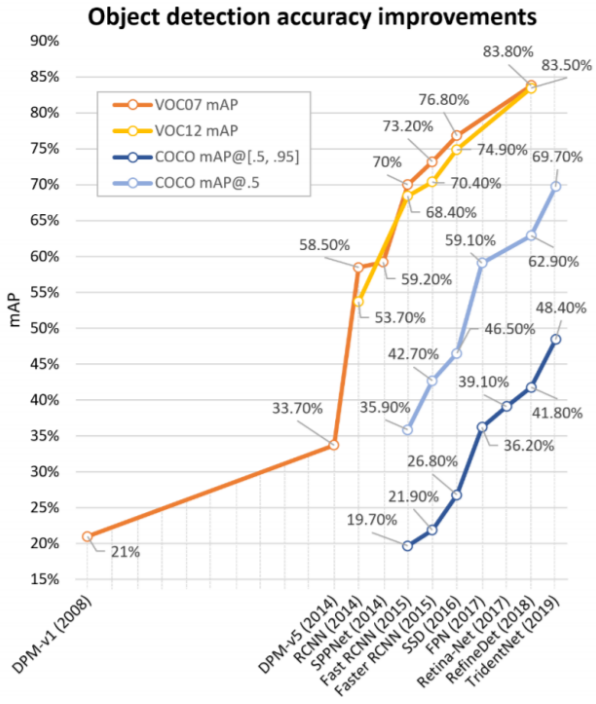
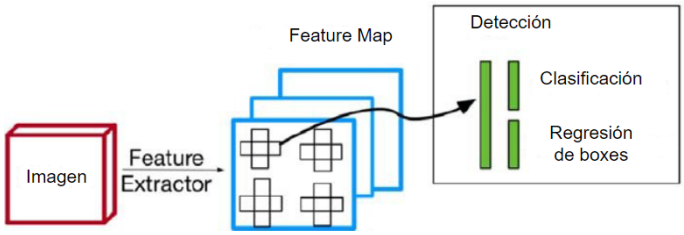
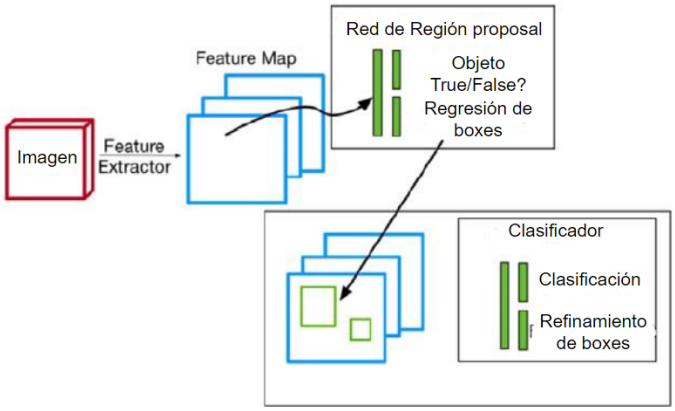


Figura 3.1: Mejoras en exactitud en modelos de detección de objetos. Fuente: [13]

Al producirse un salto tan cuantioso en el desempeño de los modelos de detección de objetos (ver Figura 3.1), mútiples redes convolucionales se generaron a continuación, definiendo dos corrientes simultáneas, según las técnicas de *deep learning* aplicadas en estas: *two-stage detector* (Figura 3.2(b)) y *one-stage detector* (Figura 3.2(a)). La diferencia entre estos métodos está en que el *two-stage detector* consta de dos etapas para detectar un objeto: En la primera etapa se buscan todos los potenciales objetos presentes en la escena y en la segunda, se clasifica el objeto y se aproxima su tamaño. Los *one-stage detector* emplean diversos métodos para realizar este proceso en un sólo paso. Esto las hace incurrir en un *tradeoff*: si bien estos detectan más rápido, su desempeño generalmente es peor que los *two-stage*.



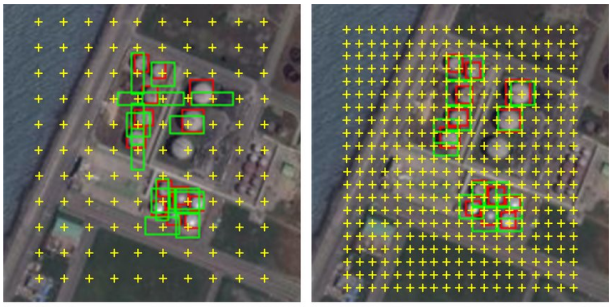
(a) Método *one-stage*



(b) Método *two-stage*

Figura 3.2: Arquitecturas de los métodos[19]

Otra clasificación de detectores se define respecto al método utilizado para inferir la posición de los objetos, este puede ser *anchor-based* o *keypoint-based*. Por un lado, el método *anchor-based* consiste en dividir la imagen en una grilla. Esta grilla define los puntos llamados "anclas" donde, a varias escalas, se prueba a clasificar el objeto que se encuentra sobre esta "ancla", este método realiza una búsqueda exhaustiva sobre toda la imagen (Ver Figura 3.3 (a)). Por otro lado, el método *keypoint-based* analiza ciertas características de la imagen como bordes, iluminación y colores para inferir un punto que describa al objeto, como su centro o esquinas y este será el *keypoint*. Una vez definidos los *keypoints* se infiere el área que contiene al objeto en su totalidad y se procede a clasificar el objeto en cuestión (Ver Figura 3.3 (b)).



(a) *Anchor-based*

(b) *Keypoint-based*

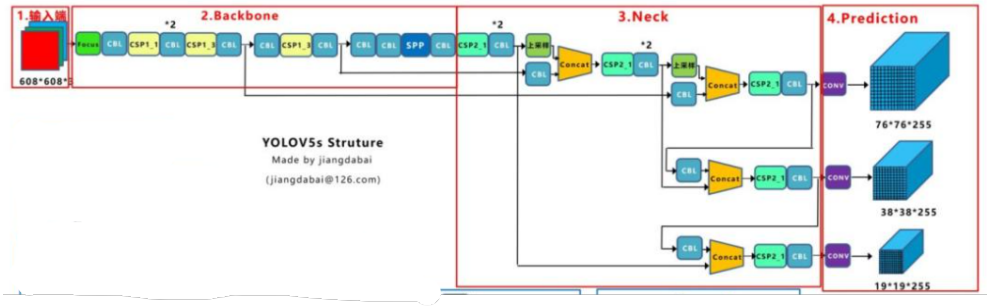
Figura 3.3: Comparación métodos *anchor-based* vs *keypoint-based*.

Para un problema específico, el detector a usar queda definido por los requerimientos y restricciones que definen al problema, es decir, si se da énfasis a la velocidad o a la precisión de la detección, ya que generalmente, al incrementar uno de estos, el otro tiende a disminuir.

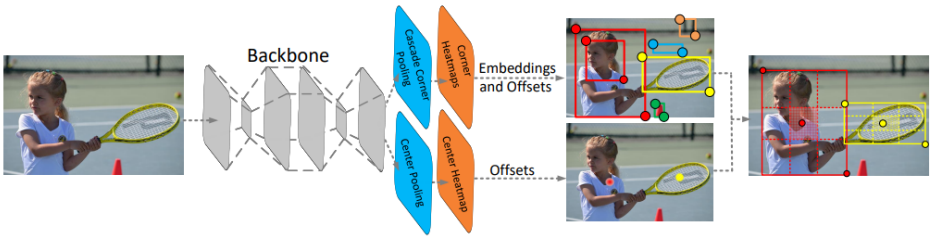
En este proyecto se realizan pruebas con dos redes de detección de objetos del estado del arte actual:

- CenterNet[4]: Para los módulos que contemplan detección de peatones, como el de distanciamiento social y el de zonas restringidas, se decide utilizar la red CenterNet para experimentar con esta arquitectura que plantea un nuevo *approach* en el ámbito de la detección de objetos, obteniendo resultados iguales o mejores[20] que otras redes del estado del arte como YOLOv3[21]. De esta manera, será posible comparar los módulos obtenidos con otras investigaciones y evaluar si el rendimiento es o no competitivo.

- YOLOv5[5]: Se decide utilizar esta arquitectura en los módulos donde es necesario un entrenamiento de la red, estos corresponden al detector de uso de mascarillas y uso de elementos de protección personal. Esta decisión se debe a que el tiempo de entrenamiento requerido por esta red es mucho menor que CenterNet, e incluso menor que redes como YOLOv3, alcanzado desempeños similares[22].



(a) YOLOv5



(b) CenterNet

Figura 3.4: Estructuras de las redes YOLOv5 y CenterNet

YOLOv5, mostrada en la Figura 3.4 (a), publicada en junio de 2020, es una red de tipo *one-stage anchor-based*. Esta es una iteración de una de las redes más populares debido a su rápida velocidad de inferencia y buen desempeño. CenterNet, por otro lado, creada en 2019 por estudiantes de *University of Chinese Academy of Sciences* y *University of Oxford*, corresponde a una red de tipo *one-stage* y *keypoint-based*. CenterNet ha sobresalido en el ámbito de detección gracias a mantener una rápida velocidad de inferencia mientras obtiene un desempeño de detección alto, incluso comparable con los detectores tipo *two-stage*.

3.1.1. YOLOv5

La primera red YOLO surge del paper publicado en 2015 por Joseph Redmon, *"You Only Look Once: Unified, Real-Time Object Detection"*[23]. Esta se popularizó rapidamente por su alta exactitud en las predicciones y su capacidad de correr en tiempo real.

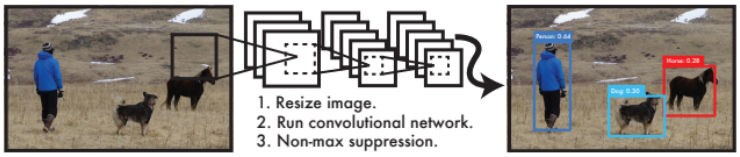


Figura 3.5: Sistema YOLO

Inspirada en la arquitectura de GoogLeNet[24], YOLO comienza dividiendo la imagen en una grilla, donde cada celda, o ancla, se encarga de analizar si existe o no un objeto en su interior. Para esto, cada celda genera múltiples tamaños de potenciales *bounding boxes* y mediante un análisis de probabilidades, define con que confianza se predice un objeto de clase C en tal celda, éste proceso se resume en la Figura 3.5. Para prevenir la detección múltiple

de un mismo objeto, en caso que este ocupe más de una celda, se emplea *Non-maximal supression*, mejorando notablemente el desempeño a cambio de ciertas desventajas.

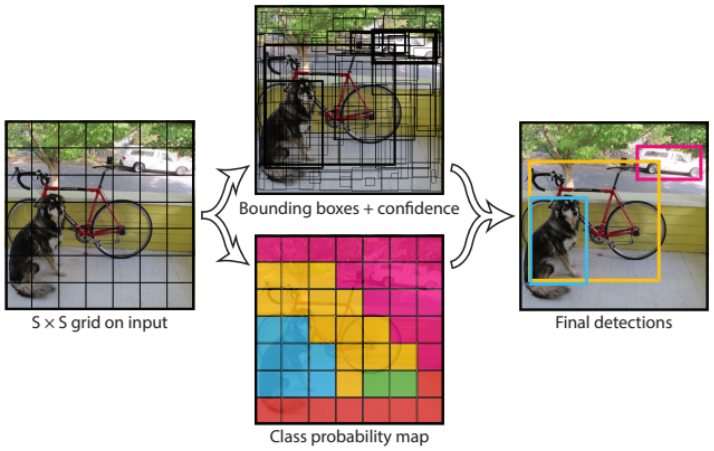


Figura 3.6: Modelo de detección

Las ventajas de YOLO sobre otros sistemas de detección (DPM, R-CNN) son atribuibles principalmente a tres factores:

- Detección como problema de regresión: Al afrontar el problema de detección como un problema de regresión, no se ocupan *pipelines*[1] complejos, aumentando la velocidad de inferencia de la red.
- Solo una red neuronal: Una sola red neuronal se entrena tanto para predecir *bounding boxes*[2] como para predecir las probabilidades de las clases.
- Información contextual: La red se entrena con imágenes completas, es decir, que contienen varios objetos en escena, por lo que adquiere información contextual de las clases.

Si bien su desempeño competía con el estado del arte cuando fue publicada, y aún lo hace, la red YOLO incurría en múltiples problemas de detección cuando trataba con objetos pequeños. En particular por la implementación de *Non-maximal supression* ya que limita las detecciones según el tamaño de las celdas, errando cuando existen dos objetos muy cercanos el uno con el otro. Es así como, dos años más tarde, el mismo autor, Joseph Redmon, publica YOLOv3[21] incorporando mejoras incrementales a la red.

El desarrollo de YOLOv3 trae consigo mejoras en cuatro ámbitos:

- Predicción de *bounding boxes*: Mejora la predicción de los *offset* de los *bounding boxes* en base al error al cuadrado del *loss*. Tambien se ajusta un umbral para determinar cual es el *bounding box* que le corresponde a cada objeto, donde siempre se le adjudica un solo *box* a cada objeto.

[1] Serie de pasos a seguir para obtener un efectivo procesamiento de datos para cumplir un objetivo específico.
[2] Figura geométrica rectangular que contiene un elemento de interés en una imagen

- Predicción de clases: Se reemplaza el uso de *softmax*[3] por múltiples clasificadores logísticos independientes. Esto permite que en cada celda, se tengan *scores* para múltiples clases, en vez de solo una, lo que trae beneficios al detectar objetos muy cercanos.

- Predicción a distintas escalas: Además de predecir *bounding boxes* en 3 escalas, se manipulan varios *feature maps*[4] extraídos en distintos segmentos de la red y mediante operaciones como upsampling o concatenación, se genera información más detallada de la imagen.

- Extractor de características: Se emplea una nueva estructura de red inspirada en YOLOv2 y Darknet-19, incorporando también técnicas de redes residuales.

En términos de desempeño, YOLOv3 en una red muy importante y disruptiva al rendir similar a otras redes *one-stage* como SSD, tal como se observa en el gráfico de la Figura 3.7, a una velocidad tres veces mayor.

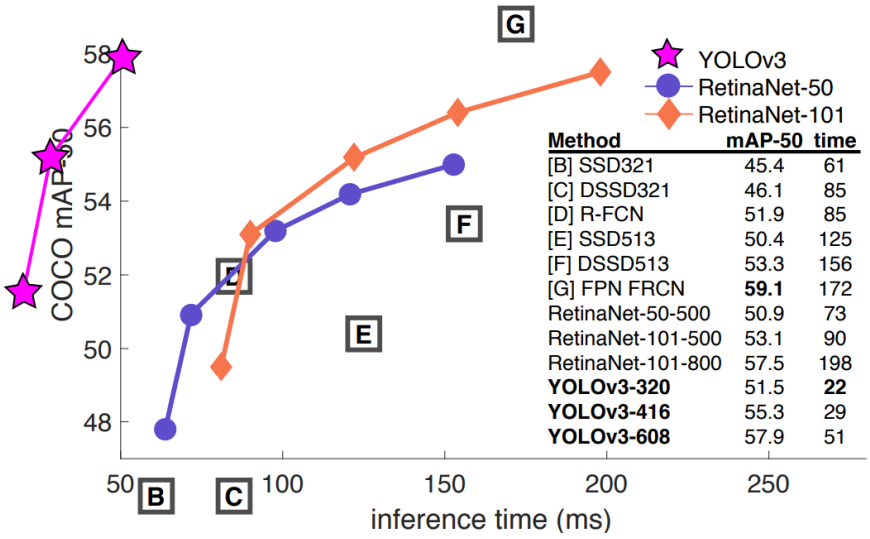


Figura 3.7: Comparación de desempeño de YOLOv3

En Junio del 2020, Glenn Jocher publicó un repositorio implementando una nueva iteración de la red: YOLOv5[5]. Este anuncio trajo consigo múltiples polémicas en diversos aspectos. Primero, Jocher no publicó un paper junto con el repositorio para detallar la investigación tras este nuevo detector. Segundo, el ingeniero y CEO de *Ultralytics*, implementó la red nativamente en PyTorch[25] a diferencia de todas las iteraciones previas basadas en Darknet. Por último, se generó una gran controversia por la publicación tan contemporánea a YOLOv4, publicada en abril del mismo año. A la fecha de escritura de este documento, esta polémica aún no cesa, en particular, debido a que ambas redes tienen un rendimiento muy similar. Sin embargo, al implementar considerables mejoras, YOLOv5 alcanza este desempeño con un tiempo de entrenamiento menor, como se observa en el gráfico de la Figura 3.9 (b).

[3] Función que convierte un vector de K valores reales a un vector de K valores reales cuya suma es igual a 1.
[4] Representación de una imagen obtenido a partir de un filtro sobre una otra, lo que permite acentuar distintas características.

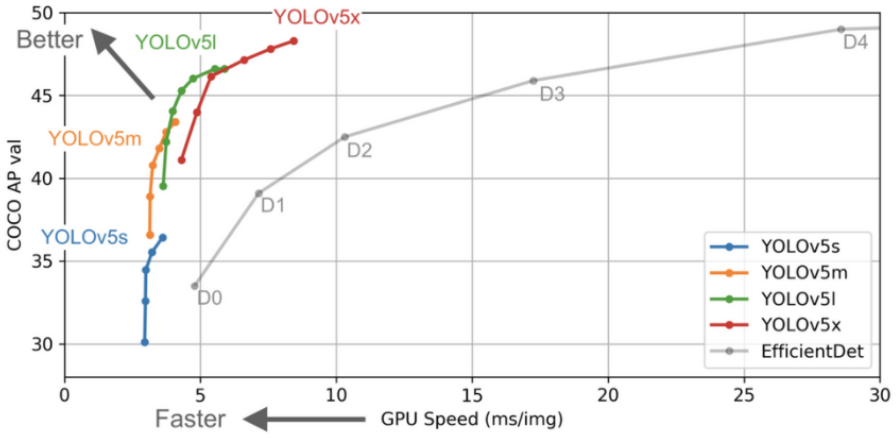
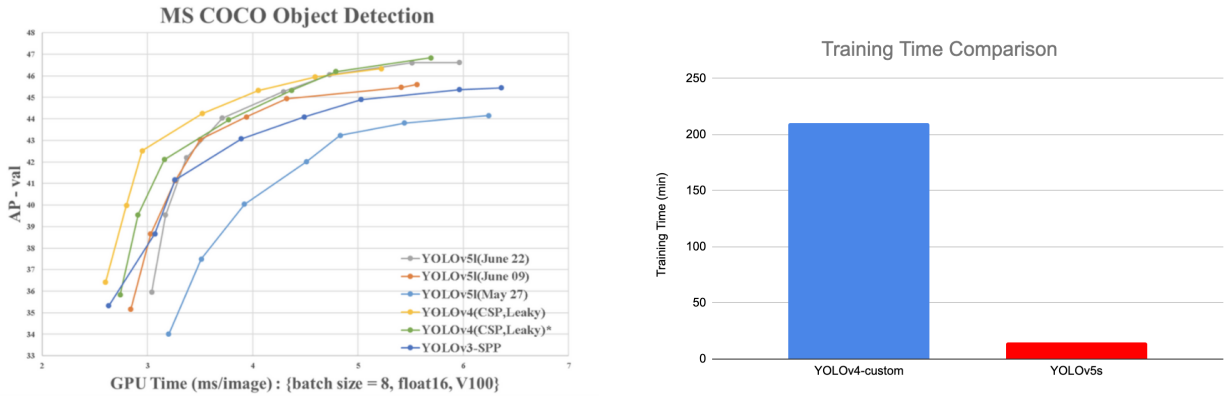


Figura 3.8: Comparación de rendimiento



(a) Comparación de rendimiento con otras iteraciones de YOLO

(b) Tiempo de entrenamiento

Figura 3.9: Comparación de rendimiento en conjunto de test de COCO dataset

En la Figura 3.9 (a), se observa un desempeño comparable e incluso mejor de parte de YOLOv4, sin embargo, WongKinYiu detalla[5] que el modelo de YOLOv4 del benchmark fue entrenado con el repositorio de YOLOv3 utilizando las técnicas de entrenamiento desarrolladas por Jocher especialmente para YOLOv5.

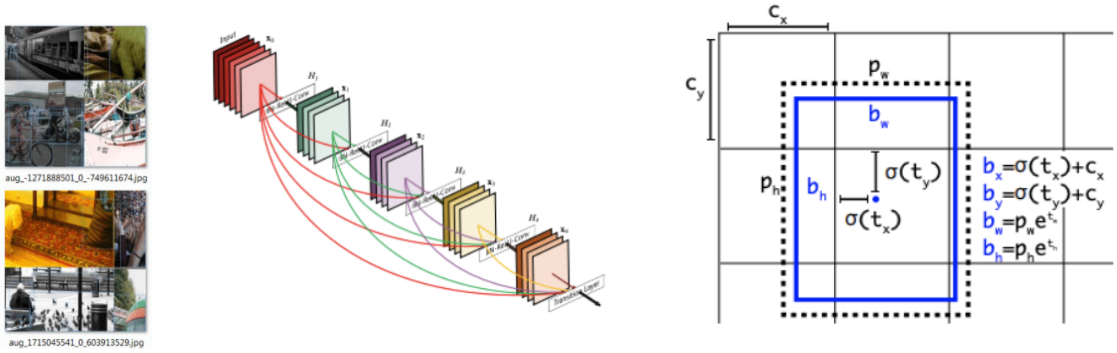
Dentro de las mejoras introducidas por el autor en esta iteración destacan las siguientes:

- Mosaic Data Augmentation: En la fase de entrenamiento se generan nuevos datos al combinar 4 imágenes de entrenamiento en una con proporciones definidas. Técnica introducida por primera vez en YOLOv4 por Glenn Jocher, el mismo autor de YOLOv5. MDA permite al modelo identificar instancias de los objetos a escalas menores que un entrenamiento normal, además de disminuir la necesidad de grandes *batches* de entrenamiento (ver Figura 3.10 (a)).

[5] Hilo en Github

- CSP Backbone: Esta red, cuyo nombre proviene de *Cross Stage Partial Networks*, alivia el problema del gradiente desvaneciente, que afecta el entrenamiento de redes profundas, a través de la implementación de conexiones entre capas no consecutivas. Basada en DenseNet, esta red es utilizada como *backbone* tanto en YOLOv4 como YOLOv5 (ver Figura 3.10 (b)).

- *Auto learning bounding box anchors*: Para algunas formas de *bounding boxes*, el método de *anchors* predefinidos tiene problemas de desempeño al tratarse de distribuciónes y ubicaciones muy distintas según las tareas. Para combatir esto, YOLOv5 genera una nueva distribucion de los *anchors* en base a los datos de entrenamiento (ver Figura 3.10 (c)).



(a) Mosaic Data Augmentation

(b) CSP backbone y conexiones entre capas no consecutivas

(c) Predicción de *bounding box* en base a *anchors*

Figura 3.10: Mejoras presentes en YOLOv5

Estas novedosas características permiten que los pesos de los entrenamientos de YOLOv5 sean alrededor de un 90 % más livianos que los pesos de YOLOv4, lo que se traduce en un menor tiempo de entrenamiento.

3.1.2. CenterNet

Esta es una red neuronal cuyo método de detección está basado en *keypoints* (a diferencia de las redes YOLO, basadas en *anchors*), es decir, en primera instancia predice puntos clave de los objetos, para luego delimitar los *bounding boxes* que los describen. Ésta nace en respuesta a los problemas presentados por la red CornerNet[26]. CornerNet es otra red basada en *keypoints*. En particular, los puntos clave utilizados por CornerNet son las esquinas de los objetos. Si bien esta red demuestra buenos resultados en general, no tiene buen rendimiento al determinar los *bounding box* para objetos pequeños.

Para combatir estos problemas, CenterNet plantea usar tripletas de puntos para cada objeto: Dos esquinas y el centro. Respecto al proceso de detección, primero se comienza infiriendo las esquinas de los objetos, para posteriormente verificar en la detección si es que existe un punto central contenido en el área delimitada por los puntos esquina previamente encontrados. Otra mejora implementada es la incorporación de dos módulos para mejorar la

detección de las esquinas y centros: *Center Pooling* y *Cascade Corner Pooling*.

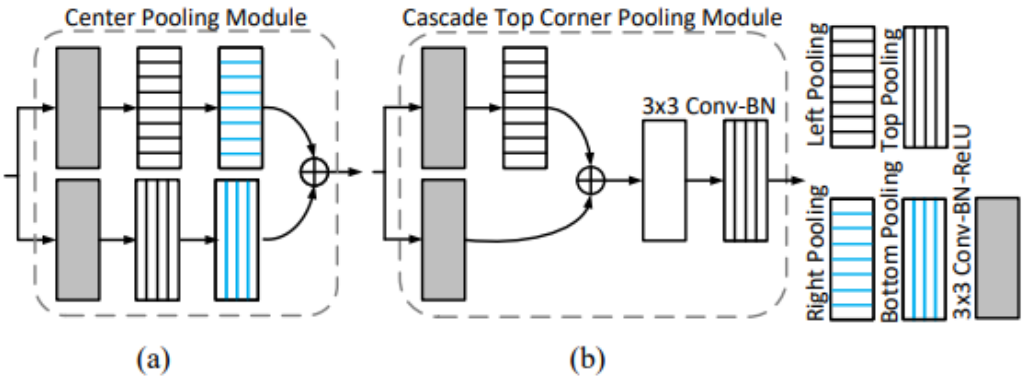


Figura 3.11: Estructura del módulo Center Pooling (a) y del Cascade Corner Pooling (b).[4]

Center Pooling, presente en la Figura 3.11 (a), es un algoritmo centrado en encontrar los puntos máximos de los objetos en los ejes horizontales y verticales para luego calcular el centro del objeto. Mientras que *Cascade Corner Pooling*, Figura 3.11 (b), es una mejora al módulo de Corner Pooling desarrollado en CornerNet que consiste en una técnica para identificar las esquinas de los objetos que usualmente se encuentran fuera de estos, esto se logra al ubicar los máximos del objeto desde afuera hacia adentro y de adentro hacia afuera en los ejes horizontales y verticales, de manera que el punto de esquina no dependa sólo de los bordes del objeto, sino también de la información interna.

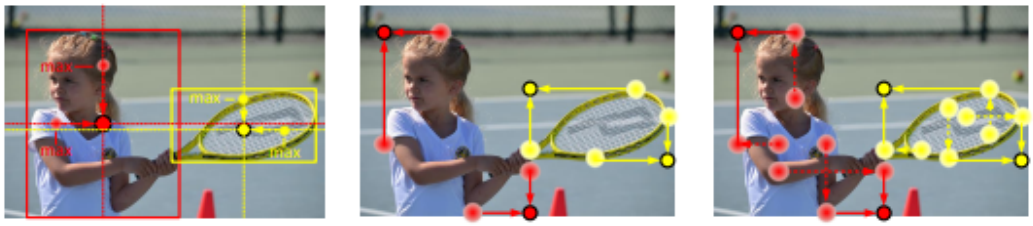


Figura 3.12: Izquierda: Center Pooling tomando los máximos del objeto para calcular el centro. Centro: Corner Pooling original tomando sólo los bordes externos del objeto para identificar esquinas. Derecha: Cascade Corner Pooling toma los máximos externos e internos del objeto para encontrar las esquinas.[4]

Los autores crearon dos arquitecturas distintas para la red CenterNet, una con Hourglass-52 como *backbone* y otra con Hourglass-104, como backbone, donde los números definen la cantidad de capas. De esta manera se definen las variantes CenterNet511-52 y CenterNet511-104 respectivamente.

Capítulo 4

Metodología

A continuación se detalla la metodología utilizada para construir cada uno de los cuatro módulos:

- Detector de distanciamiento social: Sistema que monitorea que exista una mínima distancia (definida por el usuario) entre los peatones en la imagen.
- Control de zonas restringidas: Sistema que da seguimiento a las cercanías a zonas restringidas, analizando si existen peatones cerca de estas.
- Detector de uso de mascarillas: Sistema que analiza si los rostros en la imagen se encuentran utilizando o no mascarillas.
- Detector de elementos de protección personal: Sistema que detecta si las personas en la escena utilizan casco de seguridad o no.

Para validar el funcionamiento de los módulos 1 y 2, y para entrenar los módulos 3 y 4, es necesario tener acceso a datos junto con sus respectivas etiquetas *"ground truth"*. Hoy en día existen múltiples estándares de anotaciones, estos dependen del dataset y del modelo en cuestión. PascalVOC y COCO son dos de los estándares de formateo de etiquetas más usados hoy en día. A continuación se mencionan algunas diferencias entre estos:

Anotaciones de datasets y Roboflow

- PascalVOC utiliza un archivo XML por cada imagen del dataset, mientras COCO almacena las etiquetas en un único archivo de tipo JSON para todo el conjunto de datos.
- PascalVOC describe los *bounding box* con los siguientes puntos: *(xmin-top left, ymin-top left,xmax-bottom right, ymax-bottom right)*. COCO, en cambio, usa: *(x-top left, y-top left, width, height)*.

Para el trabajo de la red CenterNet (módulos 1 y 2) se utilizan anotaciones de tipo COCO. Mientras que para la red YOLOv5 (módulos 3 y 4) se utilizan anotaciones de tipo YOLO Darknet, donde se describen las etiquetas en un archivo TXT para cada imagen. En este proyecto se utilizó la plataforma de visión computacional *Roboflow* para convertir los formatos de anotaciones según los requerimientos de los modelos.

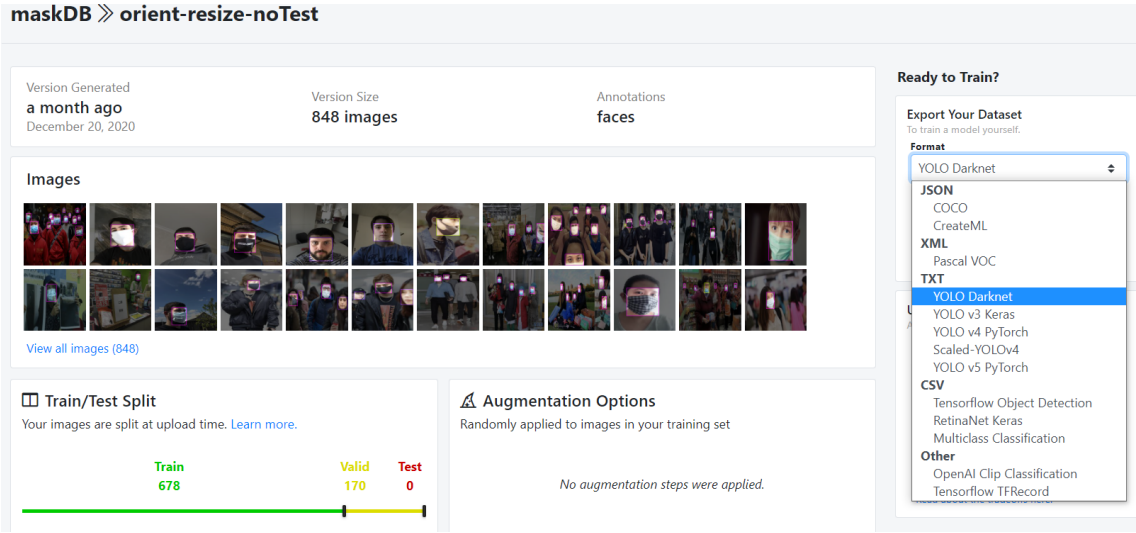


Figura 4.1: Plataforma Roboflow

4.1. Módulo 1: Detector de Distanciamiento Social

4.1.1. Objetos a detectar y metodología de los detectores

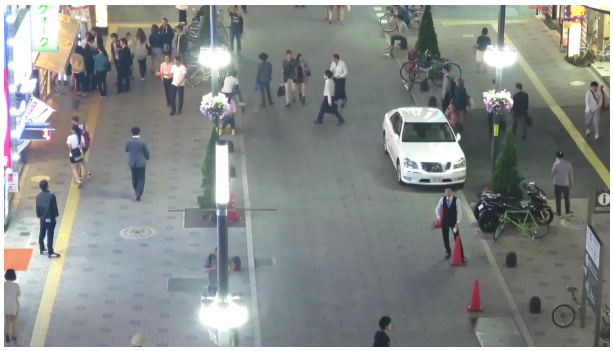
Para la detección de personas se emplea una arquitectura de red de tipo CenterNet. Una vez detectados, es necesario generar una heurística que define las reglas de las detecciones de transgresión al distanciamiento social.

En este caso, el objeto a detectar son personas. Una vez detectadas son mapeadas en una vista de dos dimensiones denominada *birdview*. Esto permite un mejor análisis de las distancias dentro de la zona de interés. En esta representación se verificará si las distancias entre los individuos supera o no un umbral definido, lo que indicará si se gatilla o no una detección de infracción al distanciamiento social.

4.1.2. Bases de Datos

Respecto al proceso de detección, al haber sido entrenada con COCO Dataset[27], la red es capaz de detectar personas. Este dataset, distinguido en el ámbito de detección de objetos, contiene más de 120.000 imágenes con un total de más de 880.000 anotaciones, sólo para la clase persona contiene alrededor de 66.000 imágenes.

Por otro lado, para el proceso de evaluación, se utilizaron dos dataset: el primero corresponde a videos obtenidos de PSINet, Figura 4.2 (c y d), los cuales apuntan a distintas zonas de las minas como casinos, barrio cívico o accesos. Estos debieron ser etiquetados manualmente para obtener el *ground truth*, por lo que se utilizó la herramienta LabelMe[28] para llevar a cabo esta tarea. El segundo dataset utilizado es el Multiple Object Tracking Benchmark Dataset (MOT)[29], Figura 4.2 (a y b), el cual se conforma de una colección de videos junto con sus etiquetas de *Ground Truth*.



(a) MOT video 1

(b) MOT video 2

(c) PSINet video 1

(d) PSINet video 2

Figura 4.2: Videos disponibles

4.1.3. Arquitectura e Implementación

La estructura de este módulo se conforma por tres sistemas que trabajan secuencialmente: Detección, Tracking y Distanciamiento. De esta manera, el objetivo final es dividido en subobjetivos delegados a sistemas distintos, simplificando así su implementación.

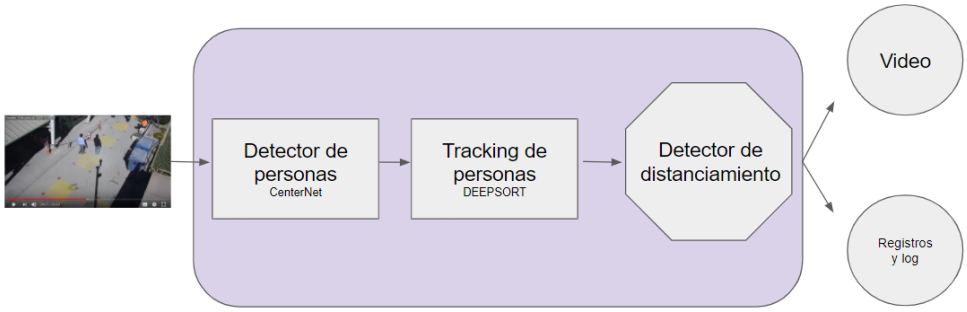


Figura 4.3: Estructura del módulo de Distanciamiento Social

4.1.3.1. Parámetros definidos por el usuario

Al ejecutar el módulo, se muestra el primer cuadro del video a procesar y se esperan algunos inputs de parte del usuario para calibrar el detector. Mediante clicks con el *mouse*

el usuario debe definir los siguientes parámetros:

- Región de interés (ROI): Corresponde a la región de interés en la escena, donde se centrará la detección del distanciamiento social. Esta debe ser definida para ajustar las distancias según la perspectiva que posee la cámara. Esta se marca clickeando las cuatro esquinas de un rectángulo en orden antihorario.

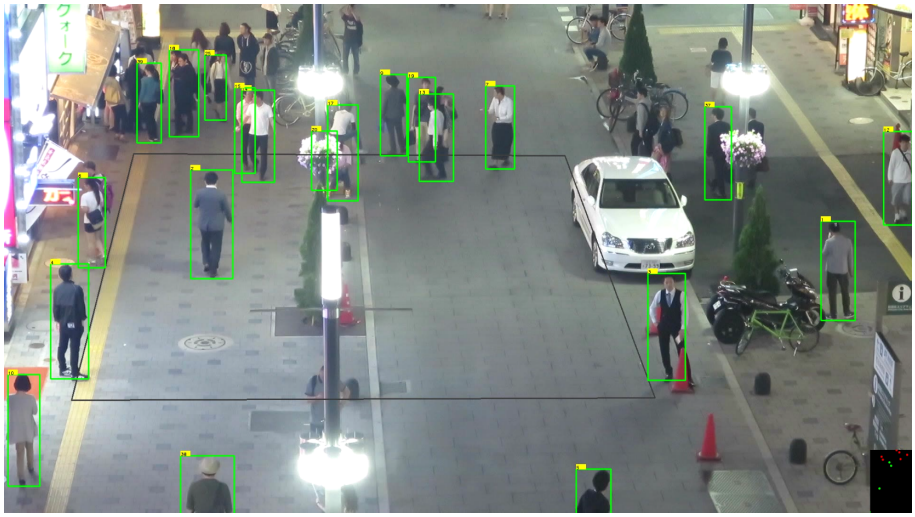


Figura 4.4: La región de interés es delimitada por un cuadrante negro y las detecciones en verde

- Umbral de distancia: Mediante tres puntos equidistantes se debe definir el umbral mínimo de distanciamiento social entre las personas. Se utilizan tres puntos en vez de dos para tener una aproximación más precisa a la distancia buscada.

4.1.3.2. Detección

El sistema de detección tiene como objetivo ubicar a las personas presentes en la escena. Para esto se basa en una red neuronal CenterNet[4] capaz de detectar personas, entre otros objetos.

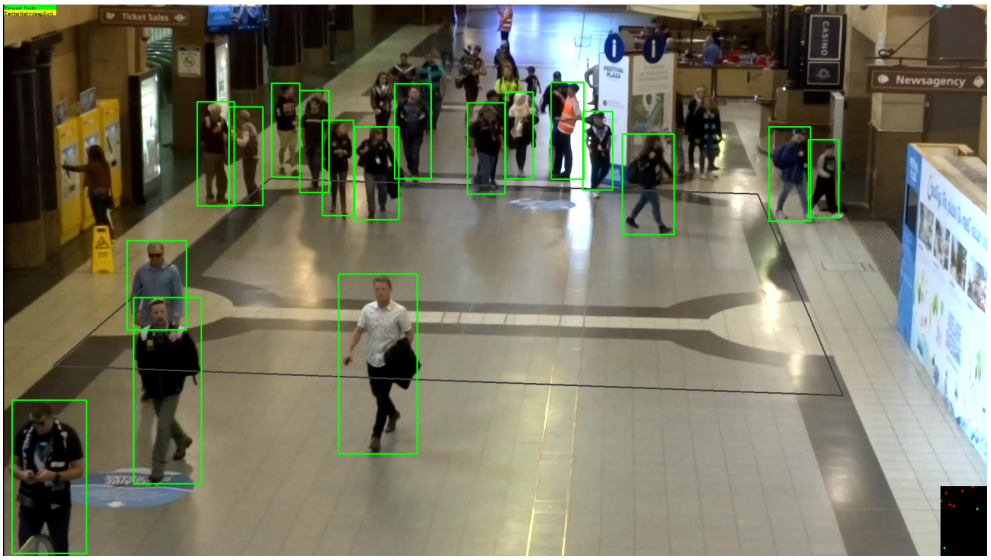


Figura 4.5: Detecciones en MOT dataset

Este primer sistema, recibe como entrada un video, para proceder a analizar cuadro por cuadro las personas en la imagen. Esta detección se realiza mediante una red CenterNet con backbone Hourglass y preentrenada con el dataset COCO[27], por lo que es capaz de detectar peatones, como se observa en la arquitectura de la Figura 4.6.

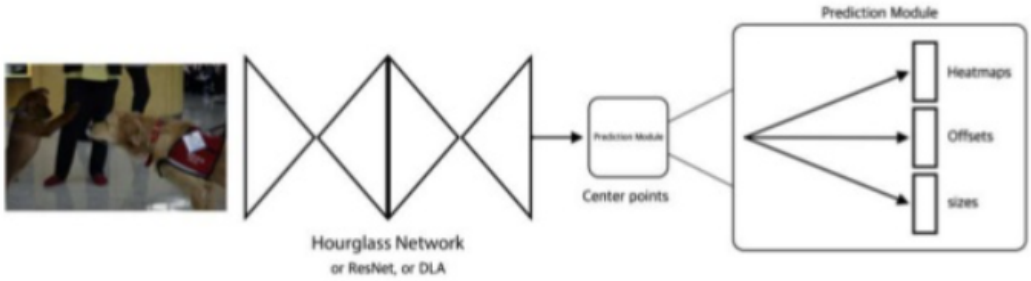


Figura 4.6: Arquitectura de la red CenterNet con bacbone Hourglass

Cada vez que pasa un cuadro por la red, esta entrega como salida una lista con *bounding boxes* de instancias de personas identificadas, y cada una con su factor de confianza asociado. Con esta información, fue necesario fijar un umbral o *threshold* para definir cuándo una detección es lo suficientemente confiable como para ser considerada. Este fue fijado en 0.2 al obtener un buen balance entre detecciones correctas y detecciones falsas, como se detalla en la Figura 5.12 en la sección de Resultados Obtenidos.

```
person 0.763 1728 458 1809 672
person 0.732 96 552 178 795
person 0.710 368 405 454 653
person 0.656 1356 568 1438 802
person 0.612 1106 978 1175 1077
person 0.606 225 400 286 596
person 0.604 1044 132 1103 310
person 0.599 793 148 849 322
person 0.597 1091 205 1152 389
person 0.582 1629 259 1698 445
person 0.571 497 134 551 301
person 0.559 1794 203 1856 390
person 0.509 544 147 600 325
person 0.506 682 279 739 408
person 0.491 262 376 325 592
```

Figura 4.7: Lista con *bounding boxes* de detecciones de un solo cuadro de video. Se indica la clase detectada, el nivel de confianza obtenido y puntos que describen el *bounding box*.

4.1.3.3. Tracking

El objetivo del módulo de tracking es incorporar información temporal a aquella información proveniente del módulo de Detección, para asociar las detecciones entre un *frame* y otro, mejorando así la calidad de las detecciones y diferenciando instancias de personas.

El tracking se lleva a cabo mediante *DeepSORT*[30], un ensamble de herramientas como filtro de Kalman, algoritmo húngaro y redes neuronales, las que, secuencialmente trabajan para conectar la información de las detecciones entre cuadros consecutivos. *DeepSORT* funciona de la siguiente manera:

$$d^{(1)}(i,j) = (\boldsymbol{d}_j - \boldsymbol{y}_i)^{\mathrm{T}} \boldsymbol{S}_i^{-1} (\boldsymbol{d}_j - \boldsymbol{y}_i),$$

(a) Métrica 1: Información de movimiento. Cada track i viene descrito en el espacio por (y_i, S_i) y el *bounding box* j por d_j.

$$d^{(2)}(i,j) = \min\{1 - \boldsymbol{r}_j^{\mathrm{T}} \boldsymbol{r}_k^{(i)} \mid \boldsymbol{r}_k^{(i)} \in \mathcal{R}_i\}.$$

(b) Métrica 2: r_j corresponde a la descripción de apariencia computada. Esta métrica mide la mínima distancia coseno entre el *track* i y la detección j en el espacio de apariencias.

Figura 4.8: Métricas utilizadas en *DeepSort*

Dos métricas predominan en el funcionamiento de DeepSort: La primera corresponde al cálculo de distancia de Mahalanobis entre la predicción del cuadro actual y el siguiente entregada por el filtro de Kalman, mientras la segunda se obtiene al medir la distancia coseno entre las apariencias de una instancia previa y una actual. De este modo, como se describe en las ecuaciones de la Figura 4.8, se considera tanto la información de la ruta descrita por el objeto como de la apariencia de éste para hacer un *tracking* efectivo.

$$c_{i,j} = \lambda\, d^{(1)}(i,j) + (1-\lambda) d^{(2)}(i,j)$$

(a) Se combinan ambas métricas para generar una suma ponderada que permite calcular la matriz de costos.

$$b_{i,j} = \prod_{m=1}^{2} b_{i,j}^{(m)}.$$

(b) Condición de admisibilidad a partir de umbrales definidos sobre las métricas $d^{(1)}$ y $d^{(2)}$.

Figura 4.9: Ecuaciones de *DeepSort*

Al combinar ambas métricas se genera la matriz de costos C, descrita en la Figura 4.9. Esta matriz C junto con restricciones definidas por las distancias espaciales en la escena permiten definir el algoritmo de optimización para generar el *matching* entre detecciones.

Listing 1 Matching Cascade

Input: Track indices $\mathcal{T} = \{1, \ldots, N\}$, Detection indices $\mathcal{D} = \{1, \ldots, M\}$, Maximum age $A_{\max}$

Compute cost matrix $\boldsymbol{C} = [c_{i,j}]$ using Eq. 5
Compute gate matrix $\boldsymbol{B} = [b_{i,j}]$ using Eq. 6
Initialize set of matches $\mathcal{M} \leftarrow \emptyset$
Initialize set of unmatched detections $\mathcal{U} \leftarrow \mathcal{D}$
for $n \in \{1, \ldots, A_{\max}\}$ **do**
  Select tracks by age $\mathcal{T}_n \leftarrow \{i \in \mathcal{T} \mid a_i = n\}$
  $[x_{i,j}] \leftarrow \text{min\_cost\_matching}(\boldsymbol{C}, \mathcal{T}_n, \mathcal{U})$
  $\mathcal{M} \leftarrow \mathcal{M} \cup \{(i,j) \mid b_{i,j} \cdot x_{i,j} > 0\}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{j \mid \sum_i b_{i,j} \cdot x_{i,j} > 0\}$
end for
return $\mathcal{M}, \mathcal{U}$

Figura 4.10: Algoritmo de *matching*

La variante DeepSort, a diferencia de SORT incorpora además una red neuronal para describir las apariencias y relacionar instancias estéticamente similares.

Name	Patch Size/Stride	Output Size
Conv 1	$3 \times 3/1$	$32 \times 128 \times 64$
Conv 2	$3 \times 3/1$	$32 \times 128 \times 64$
Max Pool 3	$3 \times 3/2$	$32 \times 64 \times 32$
Residual 4	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 5	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 6	$3 \times 3/2$	$64 \times 32 \times 16$
Residual 7	$3 \times 3/1$	$64 \times 32 \times 16$
Residual 8	$3 \times 3/2$	$128 \times 16 \times 8$
Residual 9	$3 \times 3/1$	$128 \times 16 \times 8$
Dense 10		128
Batch and ℓ_2 normalization		128

Figura 4.11: Arquitectura de la red neuronal descriptora de apariencias

4.1.3.4. Distanciamiento

De los módulos anteriormente descritos se obtiene la información respecto a las personas presentes en la escena. Esta corresponde a las posiciones y grado de confianza asociado a

cada detección.

El punto de referencia para cada persona serán los pies de esta, es decir, el punto medio del lado inferior de su *bounding box* asociado (ver Figura 4.12). Para ajustar las distancias entre personas según la perspectiva de la zona de interés previamente definida, se mapean las personas en un nuevo mapa, generado mediante operaciones matriciales, aplicando la función *perspectiveTransform* de la librería OpenCV [31]. Esta nueva visualización se denomina *Bird View*, al representar una vista cenital de la zona de interés.

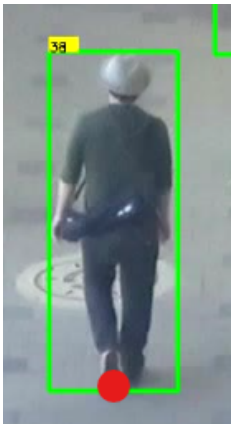


Figura 4.12: Pies de la detección: Punto medio del lado inferior del *bounding box*

Una vez obtenido el *Bird View*, como el que se observa en la Figura 4.13, se procede a determinar las distancias entre cada una de las personas de la escena. Para esto se analizan todas las combinaciones de personas detectadas y se calcula la norma euclideana para cada combinación. Si la distancia obtenida es menor al umbral previamente definido por el usuario, entonces a esta pareja de personas se le asigna un valor de *closeness* igual a 1. Por otro lado, si la distancia es mayor entonces se asigna el valor 0.

(a) Área de interés (b) Bird View

Figura 4.13: Ejemplo de detección de peatones y mapeo correspondiente en su *Bird View*

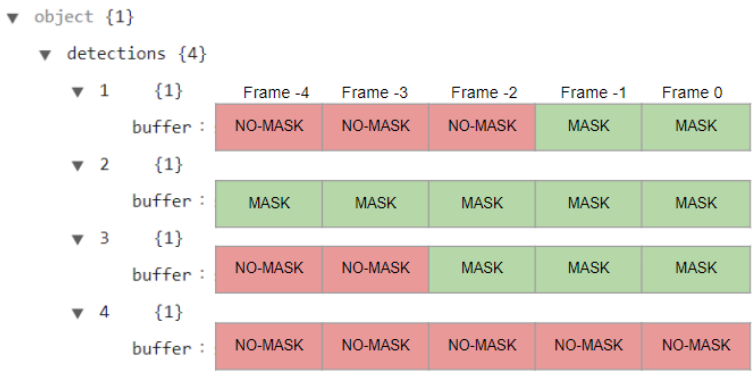


Figura 4.14: Diccionario de *buffers* de las detecciones

Para combatir las variaciones en las detecciones entre cuadros sucesivos y almacenar información temporal, se implementaron *buffers* asociados a cada pareja de detecciones. Estos se almacenan en el diccionario descrito en la Figura 4.14.

En una lista creada con la librería *deque* de *Python*, se van almacenando los valores de *closeness* asociados a una pareja en específico. Cada valor corresponde a la información de aquella pareja en un instante determinado. Es importante recalcar que esta lista es de largo fijo e igual a 51. Éste valor fue definido en base al criterio de que los videos en general son en 25/30 cuadros por segundo, por lo que la lista almacenará información de 2 segundos aproximadamente, período que permite reducir la variación excesiva en las detecciones y al mismo tiempo detectar violaciones al distanciamiento de manera rápida.

Una vez generado el *buffer* de la pareja la clasificación final asociada a esta se obtiene al analizar el valor con mayor frecuencia. Si bien en este caso se utiliza el umbral de mayor frecuencia, el umbral puede modificarse para adecuar la sensibilidad del detector a los requerimientos de la tarea o usuario.

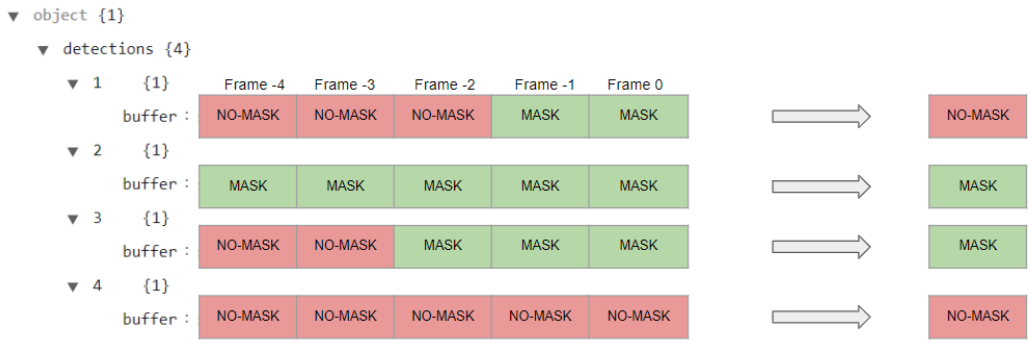


Figura 4.15: La clase de la detección se obtiene del valor de *closeness* con mayor frecuencia en su *buffer* asociado

Con la implementación de esta metodología se obtuvo un detector de distanciamiento funcional, cuyos resultados y métricas de desempeño son presentados en el capítulo a continuación.

4.2. Módulo 2: Control de Zonas Restringidas

4.2.1. Objetos a detectar y metodología de los detectores

Este módulo tiene como objetivo el análisis y detección del acercamiento de personas a zonas restringidas predefinidas por el usuario. Para llevar a cabo esto, el objeto a detectar son personas, en particular peatones, mientras que la zona con acceso limitado es definida por el usuario del programa, haciendo click con el *mouse* para delimitar esta.

Figura 4.16: Zona restringida demarcada con un cuadrante rojo

4.2.2. Bases de Datos

En este módulo se utiliza como detector la misma red CenterNet, preentrenada con COCO dataset[27], utilizada en el módulo de distanciamiento social. Para las pruebas y evaluación de desempeño se analizaron videos del dataset Multiple Object Tracking Benchmark Dataset (MOT)[29].

4.2.3. Arquitectura e Implementación

Este módulo tiene sus cimientos en la arquitectura creada para el módulo de distanciamiento. También está compuesto por tres sistemas: Detección, Tracking y Distanciamiento a zonas. Tanto Detección como Tracking se comparten con el módulo anterior, por lo que se omite su explicación en detalle. El último sistema, Distanciamiento a zonas, tiene como objetivo analizar las distancias de las detecciones y gatillar alarmas cuando sea pertinente.

4.2.3.1. Parámetros definidos por el usuario

Al ejecutar el módulo se muestra el primer cuadro del video a procesar y se esperan acciones de parte del usuario para calibrar el detector. Mediante *clicks* con el *mouse* el usuario debe definir los siguientes puntos:

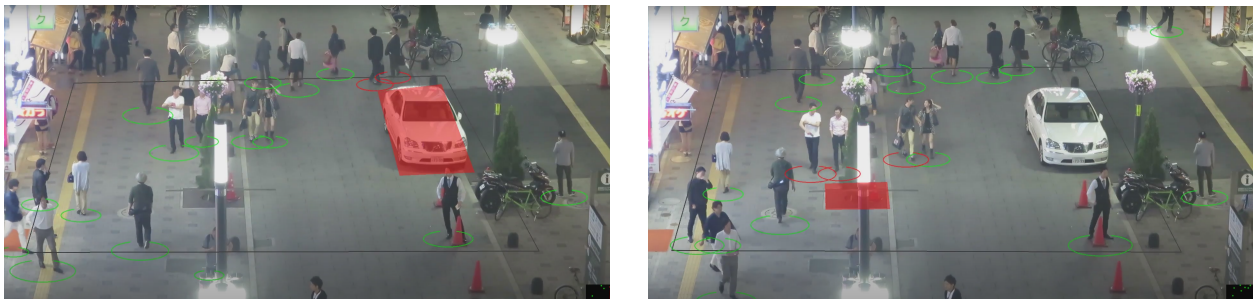
- Zona restringida: Corresponde a la zona de acceso restringido, esta se debe delimitar marcando cuatro puntos que cubran la zona en cuestión.

- Umbral de distancia: Mediante tres puntos equidistantes se debe definir el umbral mínimo de distanciamiento social entre las personas. Se utilizan tres puntos en vez de dos para tener una aproximación más precisa a la distancia buscada.

4.2.3.2. Distanciamiento a Zonas

El análisis de las distancias es similar al módulo 1 de Distanciamiento Social. Para cada detección de persona se determina su distancia a la zona restringida mediante la función *pointPolygonTest*. Esta función, perteneciente a la librería OpenCV[31], permite calcular la distancia entre un punto y el contorno de una figura, que en este caso corresponde a la zona de acceso limitado.

El factor *closeness* de cada detección de persona determina si el individuo cumple o no con la distancia mínima al sector prohibido. Se almacena este valor para cada *frame* en el *buffer* asociado a la persona, de modo que su clasificación final también se obtiene según el valor con mayor frecuencia.



(a) (b)

Figura 4.17: Zonas restringidas seleccionadas en video de dataset MOT

En el capítulo siguiente se presentan los resultados obtenidos por el módulo de Control de Zonas Restringidas.

4.3. Módulo 3: Detector de Uso de Mascarillas

4.3.1. Objetos a detectar y metodología de los detectores

El detector de uso de mascarillas se puede plantear de dos maneras distintas según detectores similares revisados en la literatura[32]. La primera consiste en detectar los rostros y mascarillas por separado, para posteriormente detectar el factor IoU entre ambos *bounding boxes*. La segunda, entrena directamente la red para detectar rostros con o sin mascarilla, sin diferenciar los objetos mascarilla y rostro. En este trabajo se optó por desarrollar el segundo detector debido a la disponibilidad de datos para entrenar los modelos.

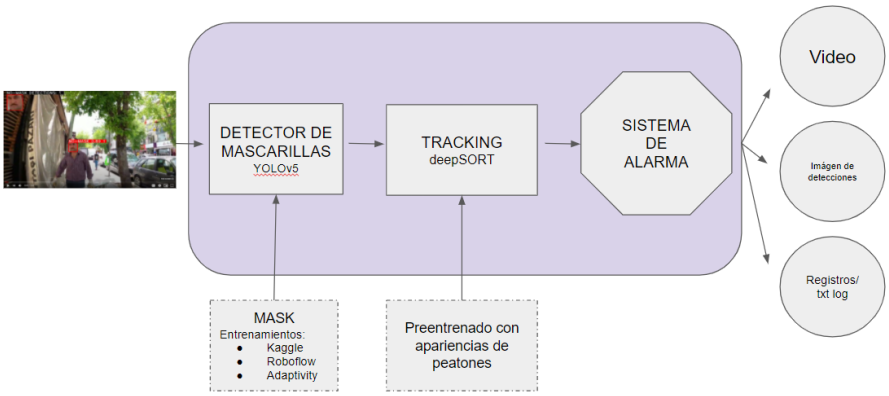


Figura 4.18: Estructura del módulo detector de mascarillas

El módulo se compone de tres partes:

- Detector de Mascarillas: Compuesto por una red neuronal YOLOv5 este modelo fue entrenado particularmente para esta tarea, siendo capaz de clasificar las clases "MASK" y "NO_MASK" según si los individuos en la escena usan o no mascarilla.
- Tracking: Para hacer *tracking* a las instancias detectadas a través de *frames* consecutivos, se incorpora *DeepSort* al igual que los módulos 1 y 2.
- Heurística o Sistema de Alarma: Consiste en un conjunto de reglas aplicado sobre las detecciones provenientes de las dos partes previas. Aquí se asocian las detecciones a los ID's de las instancias, se asocian *buffers* a las detecciones, se realiza un conteo de los rostros sin máscara y se almacenan los rostros de las infracciones detectadas.

4.3.2. Bases de Datos

Al tratarse de un tema contingente y muy reciente, al principio de esta investigación no existían bases de datos suficientes para un entrenamiento que cumpliera el objetivo buscado, por lo que en un principio la idea era generar un dataset artificial. Para esto se comenzaría con un dataset de rostros como el Labeled Faces in the Wild Database[33] y, mediante la detección de *keypoints*, se colocan mascarillas en una porción de los rostros del dataset, similar a lo realizado por Prajna Bhandary[32] como se muestra en la Figura 4.19.

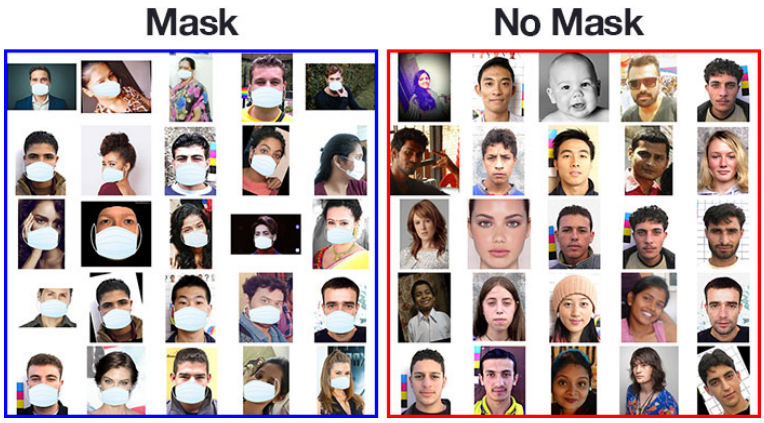
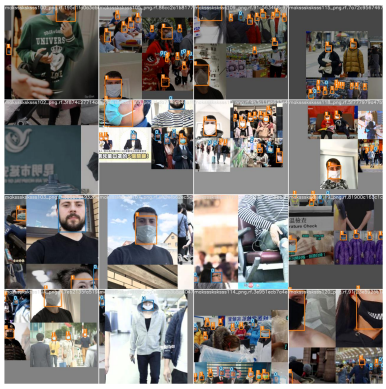


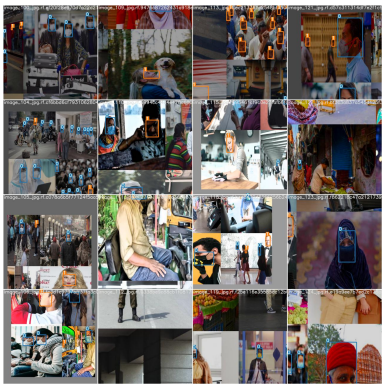
Figura 4.19: Muestras del dataset artificial de rostros con mascarilla. [34]

Sin embargo, con el paso del tiempo comenzaron a surgir diversos datasets de uso de mascarilla, lo que permitió generar múltiples entrenamientos e iterar con distintos desempeños. A continuación se detallan algunos de los conjuntos de datos utilizados:

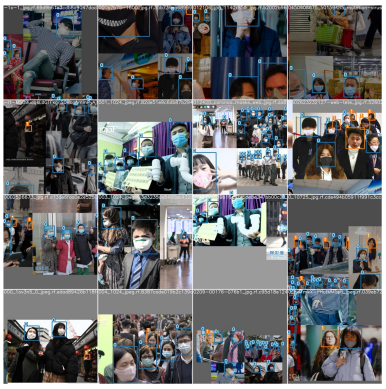
- Kaggle Face Mask Detection: 848 imágenes, 3000+ anotaciones.
- Roboflow Public Mask DB: 149 imágenes, 900+ anotaciones.
- Mask Adaptivity Dataset: 594 imágenes, 2000+ anotaciones.



(a) Datos de Kaggle Face Mask Detection Dataset



(b) Datos de Mask Adaptivity Dataset



(c) Datos de Roboflow Public Mask Dataset

Figura 4.20: Datasets utilizados

También se generó un nuevo dataset a partir de la unión de los tres datasets descritos, denominado BIG Mask Dataset, con 1591 imágenes y más de 5000 anotaciones. En la imagen de la Figura 4.21, se observa la distribución del dataset, la ubicación de los *bounding box* y sus tamaños. Para este dataset la división fue 75 % entrenamiento, 19 % validación y 6 % test.

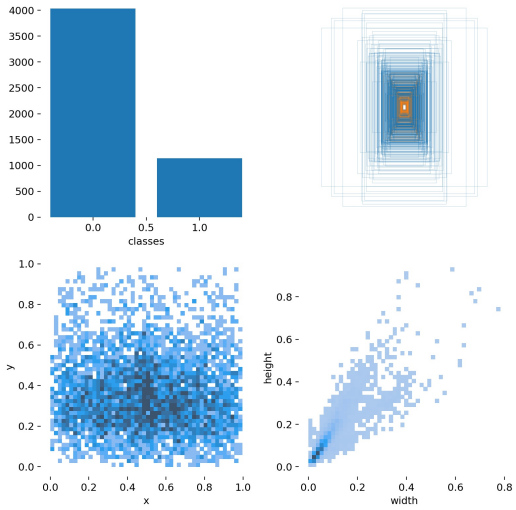


Figura 4.21: Descripción de las anotaciones en BIG dataset

4.3.3. Arquitectura e Implementación

YOLOv5 es la red seleccionada para realizar la detección de rostros, debido a su rápida inferencia y entrenamiento, junto con el buen desempeño mostrado en distintos *benchmarks*. Tras múltiples iteraciones de entrenamientos con los datasets mencionados, se decide emplear el entrenamiento sobre la base de datos BIG Mask Dataset, al tener el mejor rendimiento (los resultados se muestran más adelante). Los *bounding box* entregados por la red son posteriormente transmitidos a *DeepSort* para la incorporación de *tracking*.

Para la heurística y el sistema de alarma, se implementaron diccionarios de *buffers* similares a los de los módulos 1 y 2, pero con algunas variaciones importantes asociadas a este objetivo en particular. Los *buffers* son de largo 50 fijo, ya que en un video de 25 cuadros/segundo corresponde a 2 segundos y el umbral que determina si una detección es "NO_MASK" es de 15, lo que corresponde a 0.6 segundos en el video. Es decir, si en 50 cuadros continuos, al menos en 15 se detecta al individuo sin mascarilla, entonces se le asigna la clase "NO-MASK".

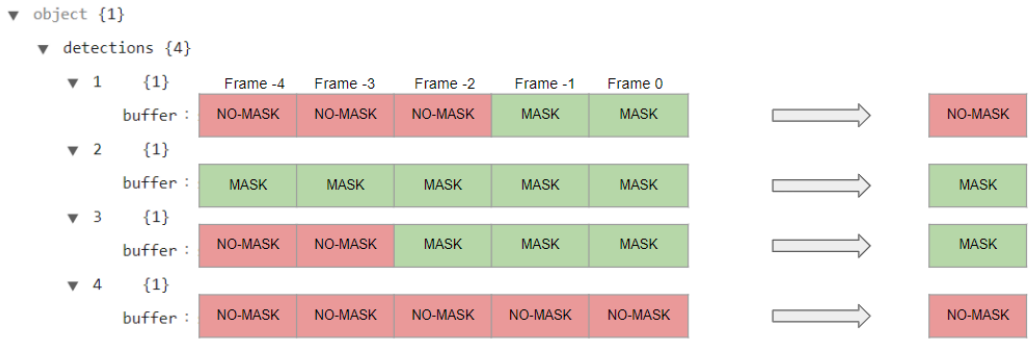


Figura 4.22: Diccionario de buffers de las detecciones

Una ejecución del módulo entrega como output el video mostrando las detecciones, la información en formato de texto y una imagen JPG con la captura de todos los rostros que fueron clasificados en algún instante como sin mascarilla. Para generar la imagen con

la colección de los rostros detectados sin mascarilla, en cada ejecución se almacena el ID de las personas detectadas sin mascarilla para así no guardar múltiples veces la detección de la misma persona. Luego, estas imágenes se van almacenando en una matriz de manera horizontal. En la sección de resultados se muestran algunos ejemplos de la imagen resultante.

4.4. Módulo 4: Detector de Elementos de Protección Personal

4.4.1. Objetos a detectar y metodología de los detectores

Este módulo tiene como objetivo la prevención de accidentes en zonas de potencial peligro, donde es obligatorio el uso de casco. Para alcanzar esto, se propone un detector de este elemento que sea capaz de analizar las escenas, almacenar información de las infracciones y alarmar en caso de ser necesario. Similar al módulo 3, este detector se compone de 3 sistemas:

- Detector de Cascos: Mediante un entrenamiento a la red YOLOv5 es capaz de clasificar las clases "HELMET" y "NO-HELMET" según si los individuos en la escena usan o no casco.
- Tracking: Implementación de *DeepSort*.
- Heurística o Sistema de Alarma: Heurística similar al módulo 3

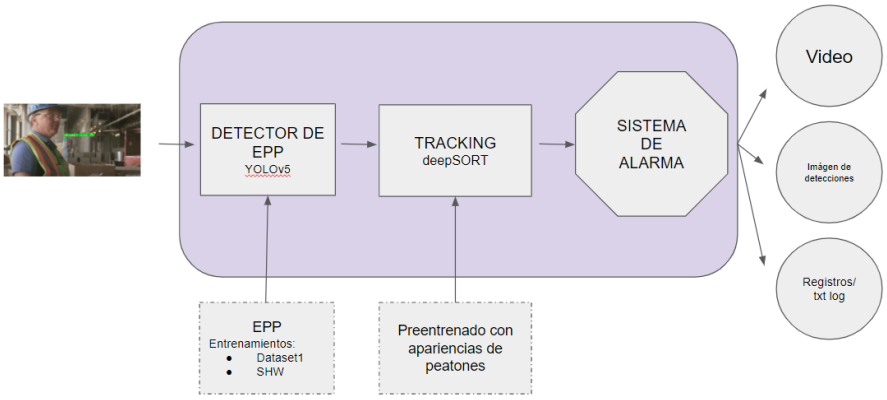
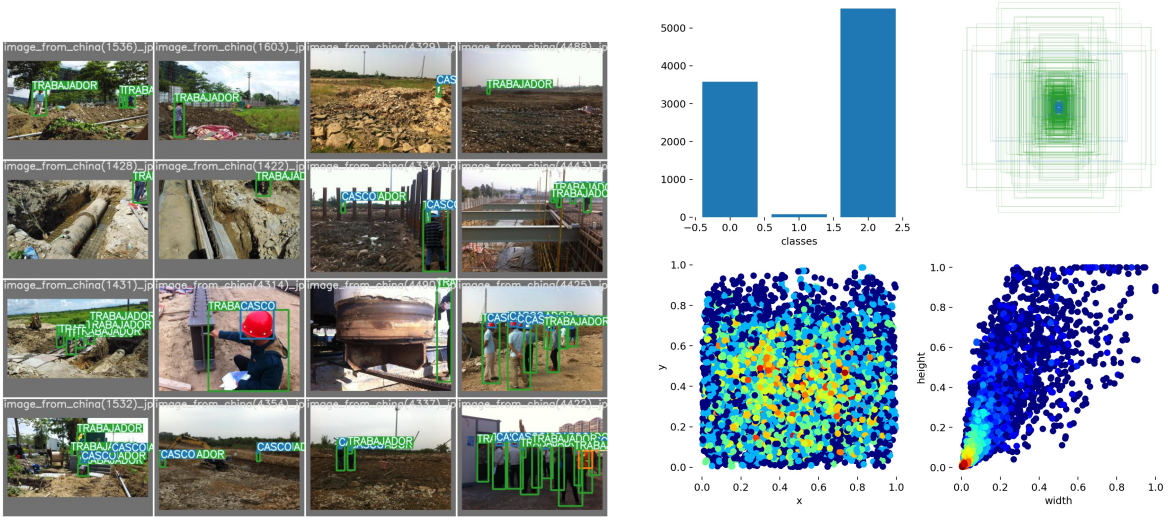


Figura 4.23: Estructura del módulo detector de EPP

4.4.2. Bases de Datos

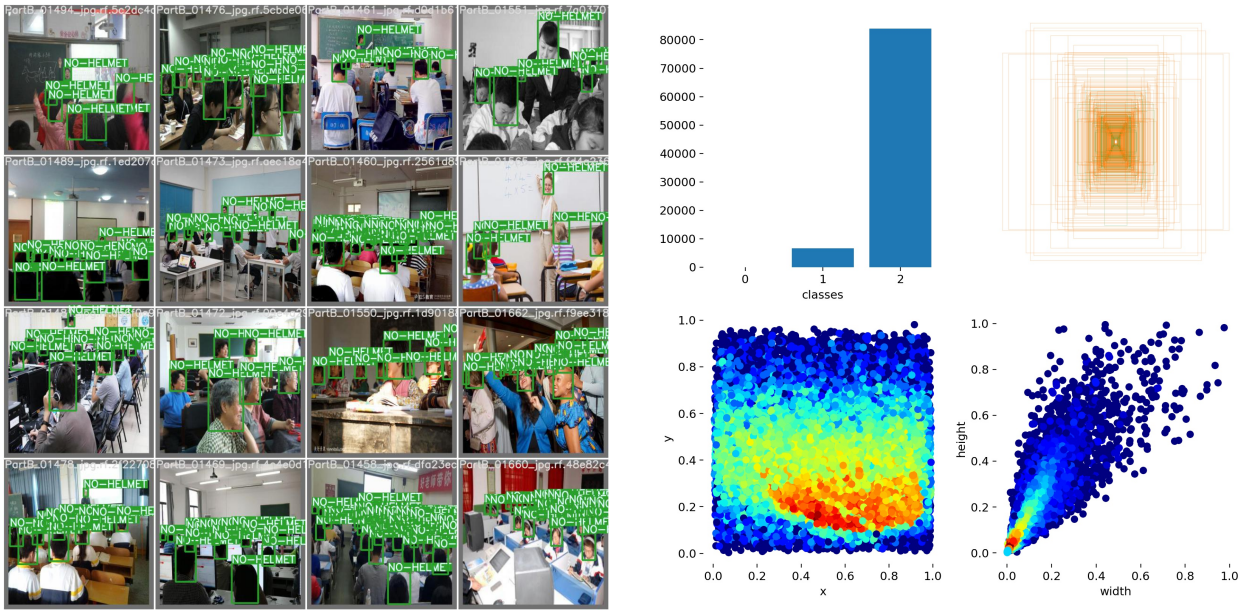
Se realizaron dos entrenamientos con dos bases de datos candidatas que afrontan el problema de detección de manera distinta. El primer entrenamiento se realizó con el dataset Pictor-v3[35], este contiene 774 imágenes con 2496 anotaciones para 3 clases: Trabajador, Casco y Chaleco. Por lo tanto para las detecciones se debía realizar una breve heurística con el fin de asociar la detección del objeto casco al objeto trabajador.



(a) Muestras (b) Distribución de etiquetas

Figura 4.24: Pictor v3

El segundo entrenamiento se efectuó con la base de datos Safety Helmet Wearing Dataset (SHW) [36]. Compuesto por 7548 imágenes, este dataset contiene más de 90.000 anotaciones de las clases “HELMET” y “NO-HELMET”.



(a) Muestras (b) Distribución de etiquetas

Figura 4.25: Safety Helmet Wearing Dataset

4.4.3. Arquitectura e Implementación

Como se mencionó, los objetos a detectar son “HELMET” y “NO-HELMET”, la red YOLOv5, es decir el sistema detector, se encarga de ubicarla en la escena y pasarle los *bounding box* detectados *DeepSort*, el segundo sistema, con el objetivo que este implemente

tracking asignándole Id's a cada una de las detecciones. Finalmente toda esta información pasa al sistema de heurística, que comparte la estructura con el módulo 3 de mascarillas, al almacenar las detecciones en *buffers*, capturar imágenes de las detecciones de "NO-HELMET" y entregar un video con las detecciones y los conteos.

Capítulo 5

Resultados Obtenidos

En este capitulo se detallan los resultados obtenidos una vez implementados los cuatro módulos y utilizados para analizar múltiples videos de prueba. Con el objetivo de cuantizar estos resultados para compararlos con otras investigaciones, y proyectos contemporáneos, primero es necesario definir algunas métricas de desempeño utilizadas.

Métricas de desempeño

Intersection over Union: IoU

Corresponde a la relación entre la intersección y la unión del *bounding box* predicho con el *ground truth*. Al fijar un umbral, permite determinar si un *bounding box* predicho corresponde a TP, FP o FN. Para evaluaciones de desempeño generalmente se fija este umbral en 0.5.

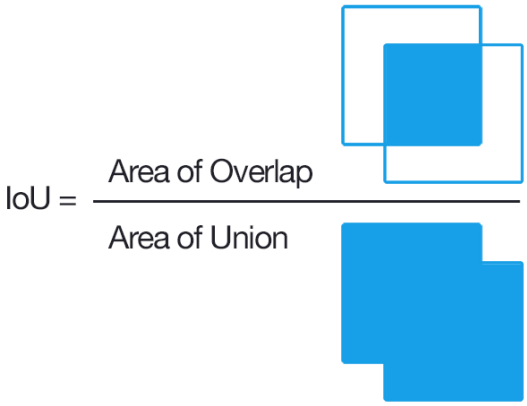


Figura 5.1: Intersection over Union

Mean Average Precision: mAP

Es una popular métrica utilizada para evaluar desempeños de modelos cuya tarea es la detección de objetos. Al ser esta una tarea de recuperación de información, la precisión se define de la relación entre las detecciones recuperadas efectivamente relevantes para la tarea sobre el total de detecciones recuperadas.

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

Figura 5.2: Precisión para tareas de recuperación de información

Al variar el valor de precisión según la cantidad de elementos relevantes en los datasets de detección de objeto, se trabaja con el *Average Precision (AP)*, el cual se obtiene al promediar la precisión cada vez que se encuentra un elemento relevante (*recall point*) en una lista única de recomendaciones.

$$AP = \frac{\sum_{k \in K} P@k \times rel(k)}{|relevantes|}$$

Figura 5.3: Average Precision

En la fórmula de la Figura 5.3, k son posiciones de ranking con elementos relevantes, P@k corresponde a la precisión en el *recall point* k y *rel(k)* es una función que indica 1 si el ítem en el ranking k es relevante, 0 si no.

Al calcular el AP sobre múltiples clases o múltiples valores de IoU se obtiene el *mean Average Precision (mAP)*

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i$$

Figura 5.4: Mean Average Precision

Curva Precision-Recall

Corresponde a la curva resultante de graficar los valores de *precision* y *recall* para un rango de valores del umbral de detección. Generalmente utilizada para evaluar modelos de detección de objetos, esta curva presenta ciertas ventajas sobre la curva ROC. Si bien la curva ROC muestra el desempeño respecto a la detección de falsos positivos y verdaderos positivos, esta se comporta de manera optimista en casos donde existe un desbalance considerable en los datos analizados. Es por esto que, para casos como el que se trabaja en este proyecto, es beneficioso tratar con la curva de *precision-recall*, al ser más sensible a las distribuciones de las clases.

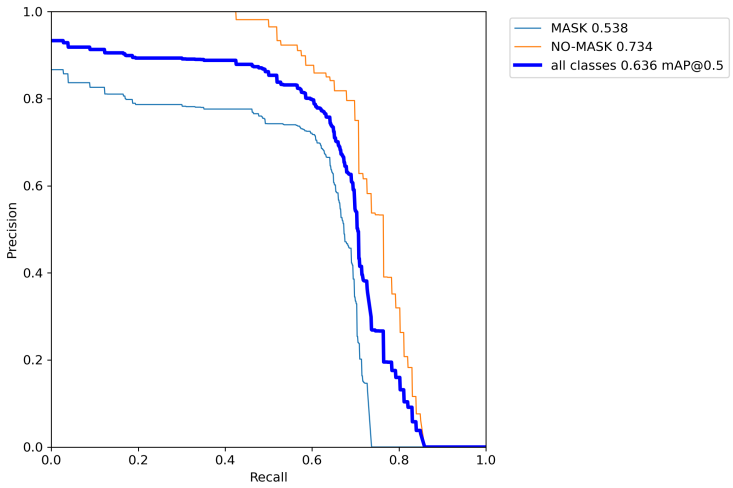


Figura 5.5: Curva *precision-recall*

Esta ventaja de la curva *precision-recall* sobre la curva ROC para datos desbalanceados, proviene directamente de que, mientras la primera extrae sus valores de dos columnas de la matriz de confusión de manera independiente (True Positive Rate (TPR) y False Positive Rate (FPR)), la segunda considera la distribución de los datos al incluir la métrica precisión y recall, considerando así la primera fila y la primera columna de la matriz de confusión, respectivamente.

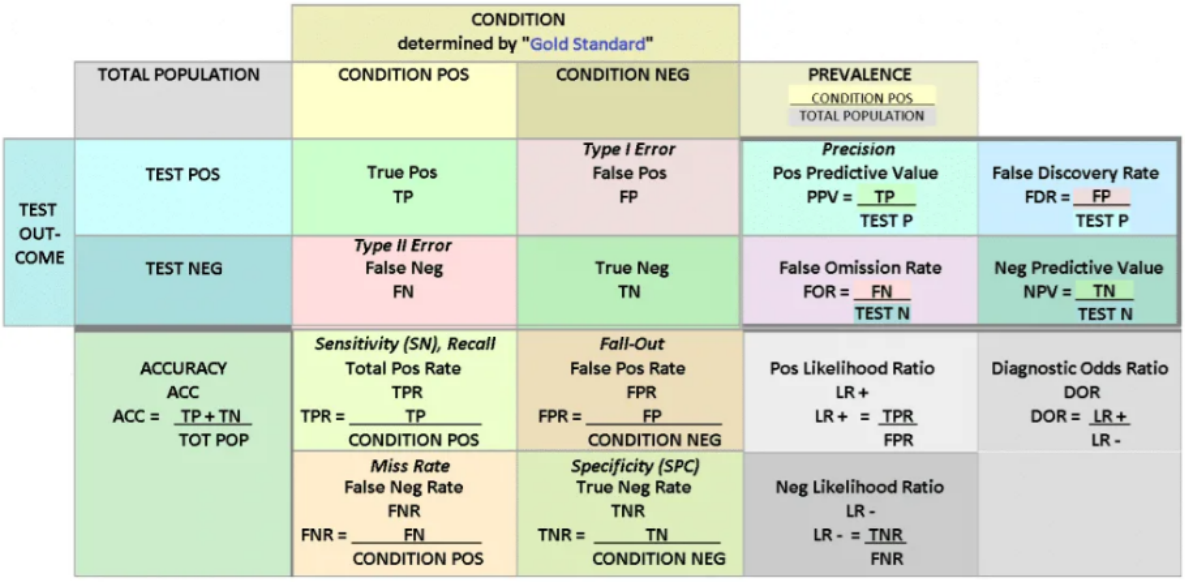


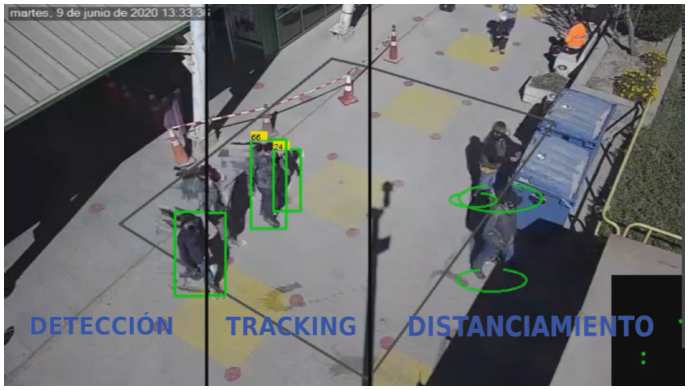
Figura 5.6: Matriz de confusión destacando las métricas TPR, FPR, *precision* y *recall*

5.1. Módulo 1: Detector de Distanciamiento Social

Como se mencionó en la metodología, este módulo se compone de tres sistemas conectados en cascada. En la Figura 5.7, correspondiente a capturas de demostraciones, se puede apreciar

como funciona cada uno de estos sistemas por separado.

(a) Base de datos MOT



(b) Base de datos PSINet

Figura 5.7: Capturas de los tres sistemas

En la columna de la izquierda, correspondiente a la salida del sistema de detección, se observa que las personas están rodeadas por una caja verde. Cada una de estas cajas o *bounding box* corresponde a una detección. En la columna central está la salida del sistema de *tracking*. Aquí se observan las mismas cajas o *bounding boxes*, pero se agrega el número en amarillo que tiene cada detección en la esquina. Este corresponde al número de instancia asociado a la detección, el cual es constante en el tiempo permitiendo hacer *tracking* de múltiples instancias de personas en simultáneo y sin información de su identidad. Finalmente, en la columna de la derecha se encuentra la salida del sistema completo. Aquí se agregó la heurística de distanciamiento la cual se ve representada con los círculos a los pies de los individuos (ver Figura 5.8 (c)), el color verde indica que no existe violación al distanciamiento, mientras que el rojo indica que sí se infringe la distancia mínima definida. Las capturas fueron obtenidas del siguiente video: Video 1.

(a) Detección (b) Instancia: Detección + *tracking* (c) Instancia con distanciamiento

Figura 5.8: Salidas de cada uno de los tres sistemas

A continuación se muestran capturas de otros videos que muestran las detecciones obtenidas por el módulo.

(a) MOT video 1

(b) MOT video 1

(c) MOT video 2

(d) MOT video 2

(e) PSINet video 1

(f) PSINet video 1

Figura 5.9: Capturas del modelo de Distanciamiento Social

En las imágenes de MOT database (Figura 5.9 (a,b,c,d)) se aprecia un mayor número de gente que en los videos de PSINet (Figura 5.9 (e,f)). En ambos, casi la totalidad de las personas tienen su círculo de detección asociado. Se evidencia, además, que las condiciones de iluminación en los primeros dos videos son mejores que en el último, al notar un menor número de sombras que podrían afectar las detecciones.

Para evaluar el desempeño del sistema se analizaron métricas de los sistemas más grandes del módulo: Detección de personas y detección de distanciamiento social. Para el primer sistema se trabaja con dos bases de datos, Multiple Object Tracking Database (MOT)[29] y una conformada por los videos otorgados por la empresa PSINet, los cuales fueron etiquetados a mano cuadro por cuadro. Al comparar las detecciones con los *ground truth* de ambas bases de datos se obtuvieron las métricas presentadas en los gráficos de la Figura 5.10. Estos comparan el desempeño de los primeros dos sistemas del módulo, es decir, las detecciones sin

tracking contra las con *tracking* mediante *DeepSort*[30].

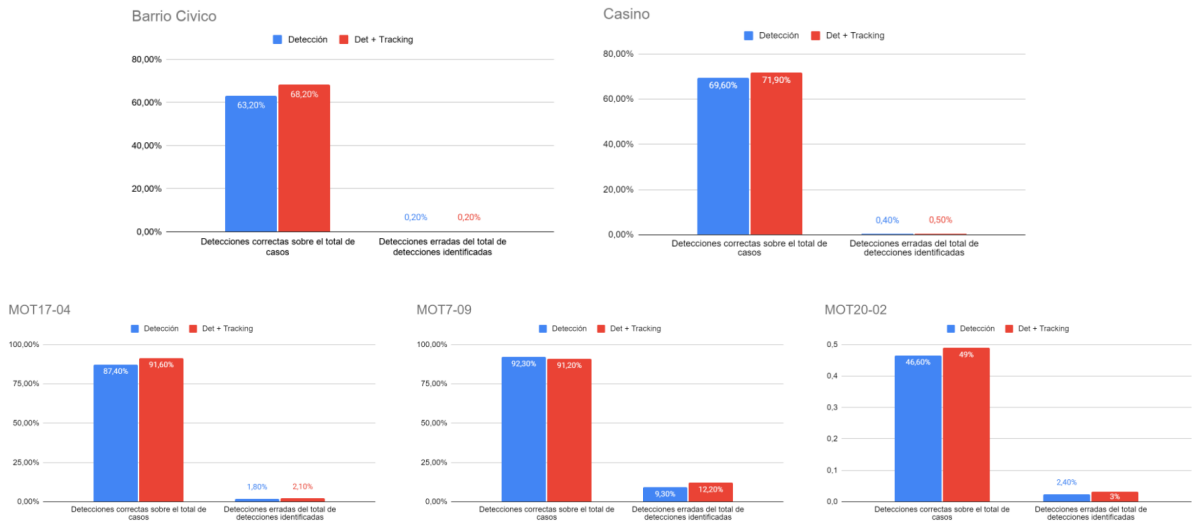


Figura 5.10: Métricas de desempeño comparando detección con y sin tracking sobre 5 videos distintos.

A continuación, en la Figura 5.11, se presentan las métricas de desempeño obtenidas para cada uno de los videos analizados, al estar evaluando un detector de objetos, el foco se centró en las detecciones correctas (*True Positive*) y las detecciones falsas (*False Positive*).

Fuente	**PSINet**		**MOT Database**		
Video	**Barrio Civico**	**Casino**	**MOT17-04**	**MOT17-09**	**MOT20-02**
Sistema	Detección + Tracking	Detección + Tracking	Detección + Tracking	Detección + Tracking	Detección + Tracking
Total de imágenes	755	2563	1049	524	1499
Total de casos	2812	6031	7934	1280	45616
Detecciones correctas (TP)	2069 (73%)	4418 (73%)	7326 (92%)	1182 (92%)	25656 (56%)
Detecciones falsas (FP)	137	410	415	370	2617

*Punto de operación (Threshold): 0.2

Figura 5.11: Métricas de desempeño del detector de distanciamiento

Las cifras que conforman las métricas mostradas son características para un punto de operación particular, el cual queda definido por múltiples parámetros que fueron ajustados en el proceso de creación del módulo. Sin embargo, existe uno que puede ser variado fácilmente según la necesidad del usuario para cumplir con las detecciones correctas o detecciones incorrectas, el umbral de detección.

El umbral o *threshold* de detección fue fijado en 0.2 para estas métricas, al considerar que entrega un buen desempeño en lo que respectan detecciones correctas y detecciones falsas. Este buen desempeño quiere decir que el detector se encuentra en un punto de equilibrio en lo que vendría a ser el *tradeoff* en el que se incurre al mover el umbral de detección, ya que uno muy alto genera pocas detecciones falsas (FP) pero pocas detecciones correctas (TP),

como ocurre en la parte hacia la derecha de los gráficos de la Figura 5.12; en comparación con un umbral más bajo el que permite un mayor número de detecciones correctas (TP) aumentando también el número de detecciones falsas (FP), hacia la izquierda en los gráficos. Algunos puntos de operación se muestran en los gráficos a continuación.

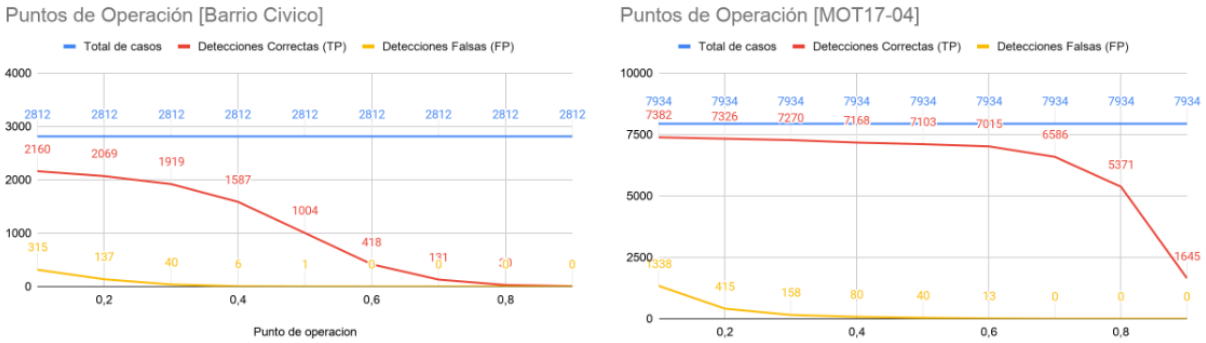
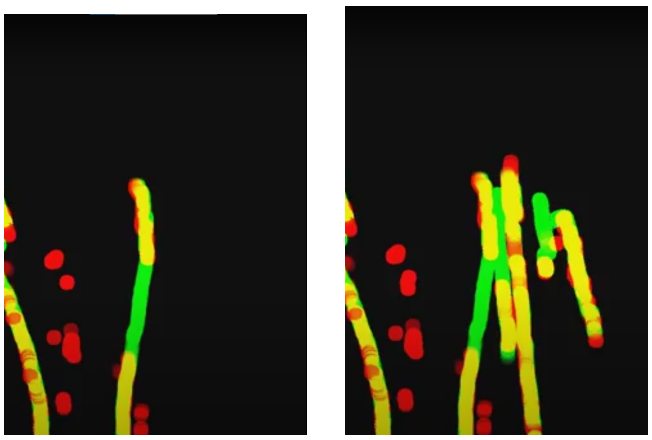


Figura 5.12: Puntos de operación sobre dos videos

Una vez analizadas las detecciones de personas, se procede a analizar las detecciones de violaciones al distanciamiento social. Para la evaluación, el *ground truth* de distanciamiento fue construido a partir del *ground truth* de detecciones, fijando un umbral de distancia que define una transgresión al distanciamento. Con el umbral en el mismo valor se ejecuta el detector sobre la escena y se comparan ambos resultados. A través del *bird view* se observan las detecciones contra las infracciones al distanciamiento.



(a) (b)

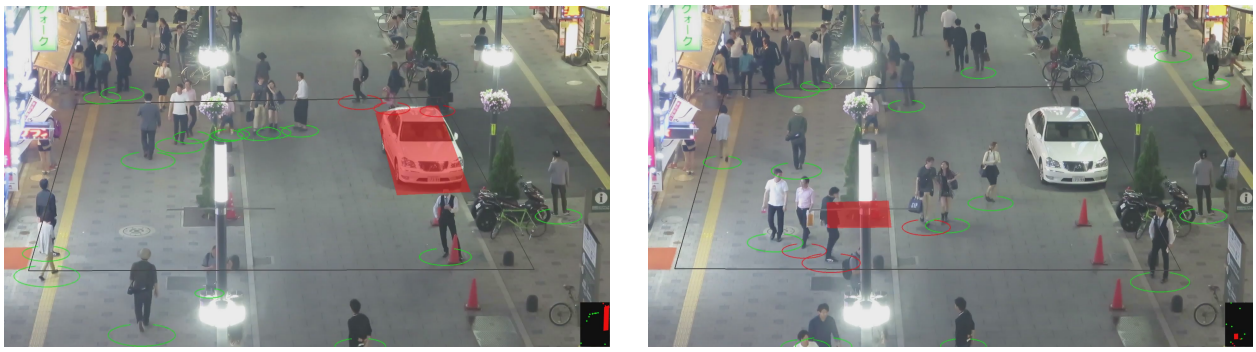
Figura 5.13: Análisis de infracciones al distanciamiento social mediante la vista *bird view*. Los puntos verdes corresponden a las infracciones *ground truth*, los rojos a las infracciones detectadas y los amarillos a los calces entre *ground truth* y detecciones.

Los puntos verdes en las imágenes de la Figura 5.13 muestran donde ocurrieron infracciones al distanciamiento, mientras que los puntos rojos muestran las infracciones detectadas. Al superponerse ambos colores, rojo y verde, generan el color amarillo, por lo que los puntos amarillos corresponden a las detecciones correctas (TP), interpretando entonces los puntos rojos como detecciones falsas (FP).

Definiendo una detección como: 1 segundo a menor cercanía de la distancia mínima. Entonces, es posible cuantizar los resultados obtenidos en ese video *benchmark*. En total existieron 306 violaciones al distanciamiento, de las cuales 221 fueron detectadas (TP) y existieron, además, 110 detecciones falsas (FP). Por lo que en resumen, de cada 10 violaciones al distanciamiento fueron detectadas 7.2.

5.2. Módulo 2: Control de Zonas Restringidas

El módulo 2 tiene como resultado un efectivo detector de acercamiento a zonas restringidas, tal como se observa en las capturas mostradas en la Figura 5.14. Se observan detecciones efectivas cuando los peatones pasan cerca del vehículo en la Figura 5.14 (a) y cerca de la planta en la Figura 5.14 (b). También es posible notar un leve retardo en cuanto a las detecciones, provocado por la cantidad de cuadros que toma superar el umbral al variar el estado en el *buffer*.



(a) (b)

Figura 5.14: Capturas de base de datos MOT con detecciones efectivas

La implementación de los *buffers* descritos previamente ocupan el rol de filtros pasa bajos aplicados a las detecciones. Esto además de disminuir las variaciones de las detecciones entre cuadros continuos añade un cierto retardo en estas, debido a que el buffer debe llenarse con el valor de *closeness* hasta superar el umbral predefinido para que la detección cambie su clase. Esto se puede observar en algunas detecciones que son levemente tardías, sin embargo, estos son parámetros ajustables acorde a las necesidades de la tarea.

5.3. Módulo 3: Detector de Uso de Mascarillas

Las métricas de los entrenamientos de la red YOLOv5 sobre los distintos datasets se encuentran resumidas en la Tabla 5.15. La base de datos BIG fue la que obtuvo los mejores valores, si se observa el índice mAP, con valores de 0.73 y 0.83 para las clases “MASK” y “NO-MASK”, respectivamente. Por lo tanto, se utilizan los pesos generados de este entrenamiento para ensamblar el módulo.

Database	Kaggle	Roboflow	Adaptivity	BIG
YOLOv5 version	small	small	small	small
Tiempo de inferencia	15 fps	15 fps	15 fps	15 fps
iou threshold	0.65	0.65	0.65	0.65
mAP mask/no-mask	0.63/0.72	0.11/0.04	0.53/0.73	0.73/0.83

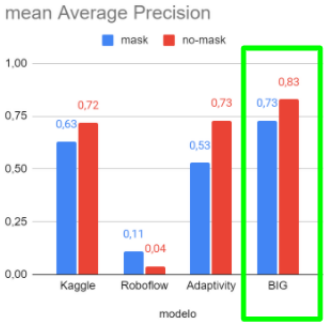


Figura 5.15: Comparación de múltiples entrenamientos para el detector de mascarillas

El entrenamiento sobre la base de datos BIG se realiza en un equipo con una GPU Tesla P100, Ubuntu 18.04, con *Python* 3.6.9 y *Pytorch* 1.6.0. Además, se define un tamaño máximo para las imágenes de 800 píxeles, considerando este un tamaño óptimo entre el detalle que tendría cada máscara, al ocupar solo una pequeña porción de la imagen, y el peso de cada imagen, que afecta el tiempo de entrenamiento; *batches* de 16 imágenes, definido por las limitaciones de RAM del equipo; y 150 épocas, número definido tras analizar que luego de las 150 épocas comienza a ocurrir *overfitting* al observar las curvas de entrenamiento. Con estos parámetros se obtuvo un entrenamiento sin *over-fitting* como se observa en las curvas de entrenamiento de la Figura 5.16.

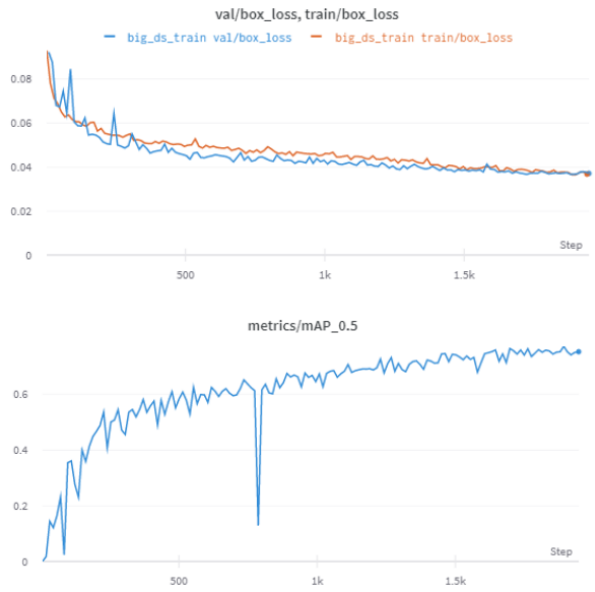


Figura 5.16: Curvas del entrenamiento con dataset BIG

La curva *precision-recall*, sobre el conjunto de test de la base de datos BIG, se muestra en la Figura 5.17, cabe destacar que de los cuatro datasets evaluados, éste fue el con mayor área bajo la curva, lo que corrobora el desempeño superior.

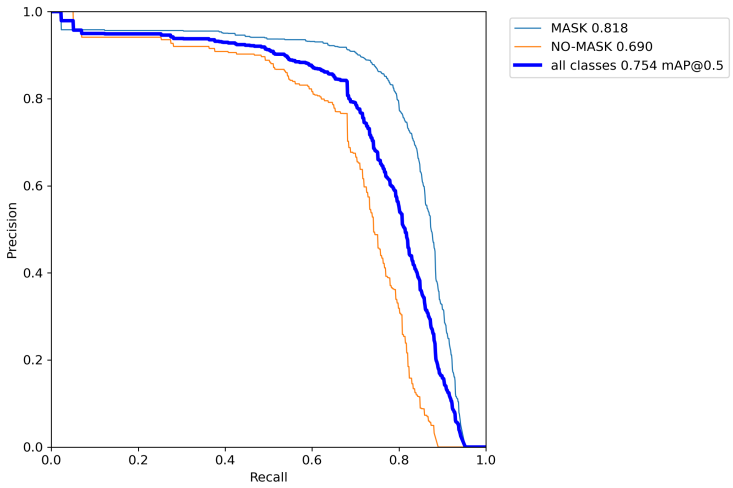
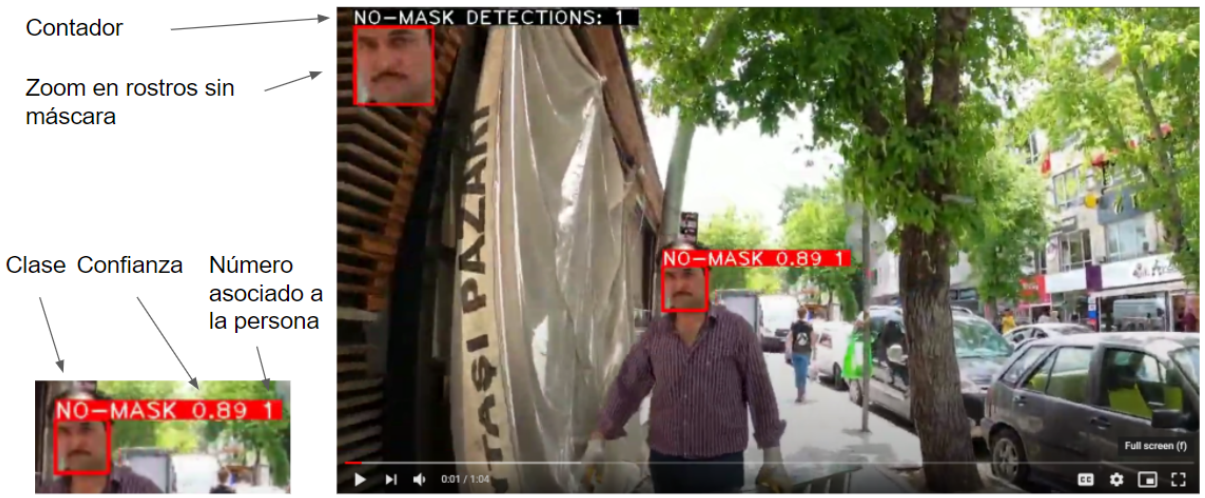
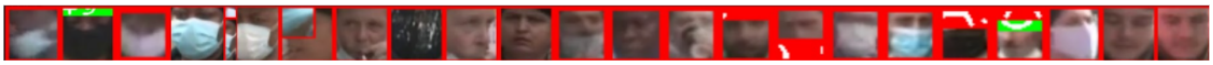


Figura 5.17: Curva *precision-recall* del detector de mascarillas

El video resultante tiene una interfaz como la que se muestra en la Figura 5.18. En la parte superior izquierda se encuentra un conteo de los rostros sin máscaras en aquel instante. Bajo el contador, se muestra un acercamiento a cada uno de los rostros detectados sin mascarilla. De éstos extractos se genera posteriormente la imagen que contiene todos los rostros detectados, como se observa en la Figura 5.18 (a). En la Figura 5.18(b) se aprecia un ejemplo del archivo resultante.



(a) Interfaz



(b) Archivo *rostros_detectados.jpg*

Figura 5.18: Salida del módulo detector de mascarillas

Las imágenes corresponden a capturas del Video 3. Más resultados obtenidos se muestran en la Figura 5.19. En estas imágenes se observa un gran número de rostros con y sin máscaras, además de los distintos tipos de mascarillas. En este dataset, si bien existen múltiples oclusiones, los videos de alta calidad, tanto en resolución como en iluminación.

(a) Video 1

(b) Video 1

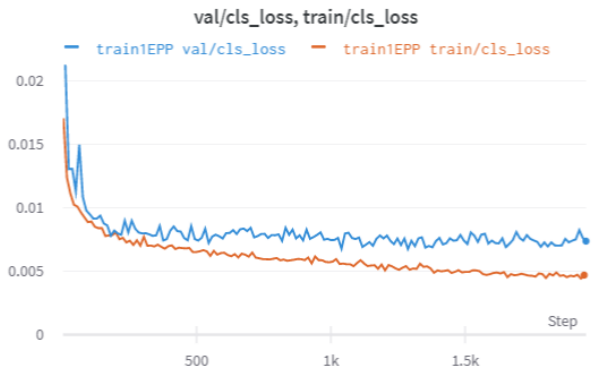
(c) Video 2

(d) Video 2

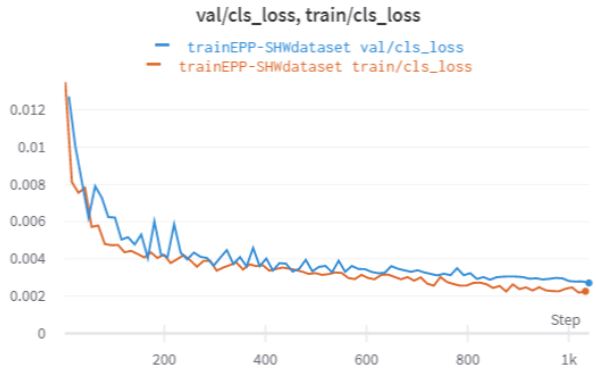
Figura 5.19: Resultados obtenidos del módulo detector de mascarillas

5.4. Módulo 4: Detector de Elementos de Protección Personal

Debido a las clases de objetos que detectan, los dos datasets propuestos para el entrenamiento de la red difieren en cuanto a como abarcan el problema. En la Figura 5.20 se observan las curvas de entrenamiento obtenidas por ambos conjuntos de datos. Ambos entrenamientos se realizaron en el mismo equipo que el módulo 3, con una GPU Tesla T100, *Python* 3.6.9, y *Pytorch* 1.6.0. Para ambos se definió un tamaño de las imagenes de 418 pixeles y batches de 16 debido a la memoria RAM disponible. Para Pictor v3 el número de épocas para obtener un buen entrenamiento sin *overfitting* fue de 150, mientras que para SHW éste valor fue de 80.



(a) Pictor v3



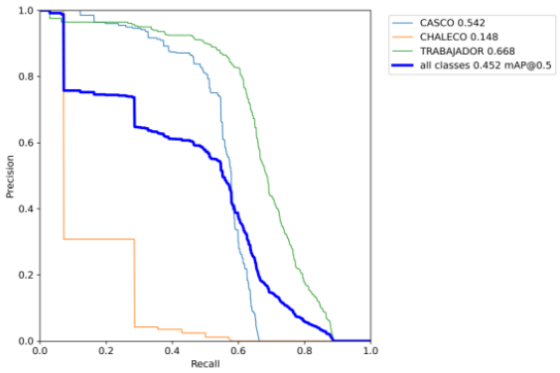
(b) Safety Helmet Wearing Dataset

Figura 5.20: Curvas de entrenamiento

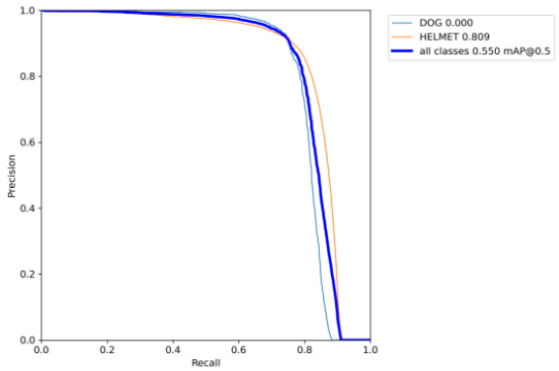
Al analizar los valores de mAP obtenidos de los entrenamientos se tiene que para el dataset Pictor v3 los valores son 0.668, 0.542 y 0.148 para las clases "TRABAJADOR", "CASCO" y "CHALECO", respectivamente. Por otro lado, del entrenamiento con los datos de SHW se obtuvo un mAP de 0.809 para la clase "CASCO".

Database	**Pictor v3**	**SHW**
YOLOv5 version	small	small
Tiempo de inferencia	15 fps	15 fps
iou threshold	0.65	0.65
mAP	casco: 0.54 chaleco: 0.14 trabajador: 0.67	casco: 0.8

(a) Tabla comparativa



(b) Curva *precision-recall* Pictor v3



(c) Curva *precision-recall* SHW

Figura 5.21: Métricas de desempeño de los modelos obtenidos

La curva *precision-recall* del dataset Pictor se muestra irregular, puesto que la curva de la clase "CHALECO" muestra un bajo desempeño. Si se comparan las curvas de ambos

conjuntos de datos se aprecia que el área bajo la curva del dataset SHW es mayor indicando un mejor rendimiento en términos generales. Por consiguiente se continúa trabajando con el entrenamiento a partir de SHW. Algunas detecciones obtenidas se muestran en la Figura 5.22.

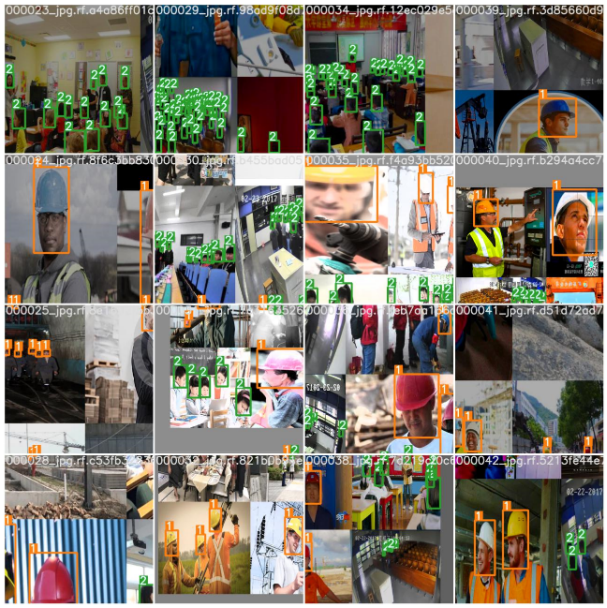


Figura 5.22: Detecciones obtenidas del dataset SHW

Una vez ensamblado el *tracking* y la heurística se realizan pruebas con videos públicos disponibles en internet. El Video 5 de la Figura 5.23d/e/f muestra un montaje en un ambiente de construcción, con una buena iluminación y resolución. Mientras el Video 4 de la Figura 5.23a/b/c muestra capturas reales de cámaras de circuito cerrado de TV de zonas de construcción. En el primero se identifican correctamente gran parte de las personas usando casco. En el segundo, existe mayor variación en las detecciones pero son detectadas todas las instancias ya sean con o sin casco.

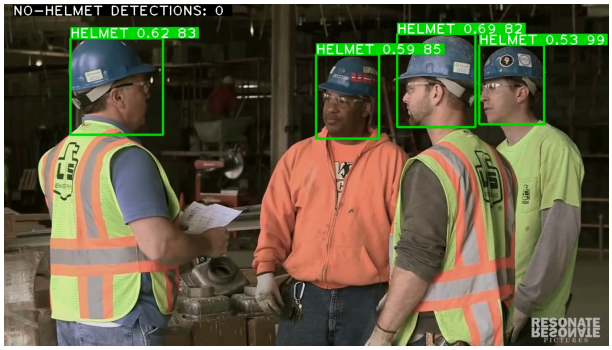
(a) Video 1

(b) Video 1

(c) Video 1

(d) Video 2

(e) Video 2



(f) Video 2

Figura 5.23: Resultados obtenidos del módulo detector de EPP

Capítulo 6

Análisis y Discusión de resultados

6.1. Módulo 1: Detector de Distanciamiento Social

En la Figura 5.10, los gráficos de barras muestran que, al incorporar *tracking* mediante DeepSort, aumenta el número de detecciones correctas en la gran mayoría de los videos analizados. Este aumento de 3 o 4 puntos es atribuible a la predicción del *bounding box* siguiente de parte del filtro de Kalman contenido en DeepSort, corrigiendo la posición de este, obteniendo así en ciertas predicciones un mayor IoU entre la predicción y el *ground truth*. Este aumento de las detecciones correctas trae consigo un leve aumento en los falsos positivos. Se tiene en promedio un aumento de 0.4 puntos. Al ser un aumento tan leve, el *tradeoff* entre estas variaciones es positivo para el sistema, por lo que *DeepSort* trae beneficios a la detección además del *tracking* de instancias correspondiente a su principal atributo.

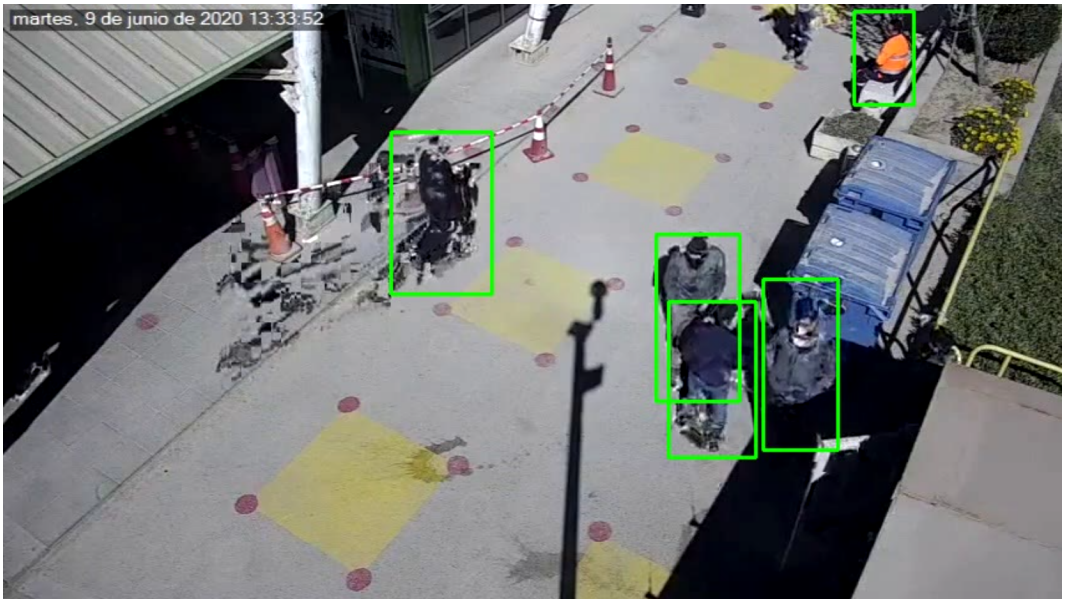


Figura 6.1: Artefactos en videos proporcionados por PSINet

Al comparar las detecciones correctas (TP) de ambas fuentes se observa que el promedio de *PSINet* es 8 puntos menor que el de *MOT Database*. Esta diferencia de rendimiento se atribuye a la calidad de los videos, en particular a las características de resolución y compresión. 854 por 480 píxeles es la resolución de uno de los videos de PSINet, mientras que

el video MOT17-04 tiene resolución 1920 por 1080. Además, en el proceso de captura de la grabación de PSINet, se incurre en una compresión para ser enviado por internet, en este proceso se generan algunos artefactos que empeoran la visualización, como se observa en la parte izquierda de la Figura 6.1, sobre los conos naranjos.

Respecto al rendimiento del detector de infracciones al distanciamiento, se obtuvo un *recall* de 72 %, detecciones en amarillo en el *Bird View* de la Figura 6.2. Una cifra alta si se compara con la limitada capacidad de un humano de analizar en tiempo real estas múltiples infracciones simultáneas o mejor aún si se compara con la ausencia de un detector. No es fácil compararse con otros detectores al no existir una definición estandarizada de cuando se incurre en una infracción al distanciamiento, ya que para este proyecto se definió como 1 segundo, sin embargo, en otros proyectos éste periodo definido puede variar.

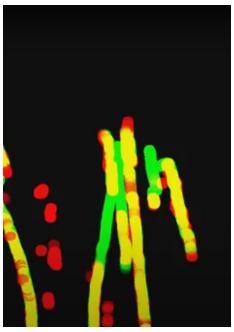


Figura 6.2: *Bird view* con el análisis de las detecciones

Los máximos valores alcanzados de detecciones correctas, o *True Positive Rate* para detección de personas, indican un buen desempeño al compararlo con otros trabajos del estado del arte[37], los que obtienen un valor de TPR igual a 90.39 %, 2 puntos bajo lo obtenido en este proyecto. Al el desempeño de detecciones de infracciones al distanciamiento social, el trabajo citado alcanza un 84 %, cuando el valor para este detector es de 72 % aprox. Si bien es un valor inferior, cabe destacar que las bases de datos de prueba difieren, lo que afecta en las métricas obtenidas. Aún así el desempeño se encuentra en un rango similar.

6.2. Módulo 2: Control de Zonas Restringidas

Al compartir arquitectura con el módulo 1 de distanciamiento social, entonces sus métricas de detección de peatones son las mismas. Para la detección de distancia a la zona restringida, al medir la distancia persona-contorno, entonces el rendimiento es igual o mejor que el módulo 1 que mide distancia persona-persona, ya que el contorno no porta consigo una probabilidad de no ser detectado al ser fijado al principio de la ejecución del programa.

Por simplicidad, la zona restringida se definió como un polígono de cuatro lados, lo que puede no ajustarse a cualquier zona, reduciendo su versatilidad, por lo que una futura implementación podría otorgar la libertad al usuario de dibujar la zona restringida o definir el número de lados que la cubren.

El rendimiento de éste módulo se encuentra en un rango entre 73 % y 93 %, ya que cada

detección depende de una persona detectada, no dos como el módulo detector de distanciamiento. Esto se debe a que la zona restringida se fija en un comienzo por el usuario. El rango de rendimiento es de 20 puntos principalmente por las diferencias calidad de los videos e iluminación de las escenas. Si se compara éste sistema con la ausencia de un sistema de seguridad, entonces la mejora es considerable. Por otro lado si se compara con un sistema de láser para monitorear zonas, entonces éste módulo es capaz de detectar el peligro antes, lo que traería ventajas a la prevención de riesgos.

6.3. Módulo 3: Detector de Uso de Mascarillas

El rendimiento superior del modelo entrenado con la base de datos BIG advertido principalmente en sus valores mAP y el mayor área bajo la curva *precision-recall* en comparación con el resto, se atribuye directamente a que la cantidad de datos que contiene es superior a los otros dataset. Este entrenamiento con más datos permite a la red una mejor generalización.

Al observar el Video 2 se aprecia que el rendimiento del detector no es perfecto, sin embargo, es efectivo al identificar gran parte de las personas que aparecen sin utilizar mascarilla. El almacenamiento de estas detecciones en una imagen facilita la comprobación del desempeño, observando que gran parte de las detecciones son, efectivamente, rostros sin mascarilla.

Estudiantes de la universidad de Hong Kong [38] desarrollaron un detector de mascaras que alcanza una precisión de 82.3 % para la clase máscaras. Si bien lo óptimo sería comparar directamente mAP contra mAP. Como se observa en la Figura 5.16, éste valor es alcanzable en ciertos puntos de operación, con un *recall* de hasta un 74 % aproximadamente, lo que indica que ambos modelos se encuentran en un rango similar en términos de precisión. En la referencia no se menciona el tiempo de inferencia, sin embargo, sería interesante comparar estos tiempos para analizar los desempeños.

6.4. Módulo 4: Detector de Elementos de Protección Personal

La elección del entrenamiento correspondió a una parte crucial en éste módulo, ya que a partir de esto se define como se abordaría el problema. En el dataset Pictor se observa un excesivo desbalance de clases, en especial de la clase chaleco al tener un submuestreo considerable. Esta es la causa de la irregularidad observada en la curva *precision-recall* y el bajo mAP mostrado para esta clase en particular. En contraste, el dataset SWH muestra un mejor desempeño al observar el área bajo la curva a pesar de tener un desbalance de clases, pero menor al de Pictor.

La curva de entrenamiento muestra una constante disminución del *loss* a través de las épocas sin un *overfitting* considerable.

Los videos de demostraciones generados muestran dos contextos muy distintos; el primero, al poseer una alta calidad y condiciones de iluminación muestra el potencial del módulo en óptimas condiciones, lo que se ve plasmado en el buen rendimiento de las detecciones.

Desde otro ángulo, en el video de la Figura 5.23a/b/c se pone a prueba el módulo en condiciones cercanas a la realidad, donde los individuos ocupan una menor porción de la escena, la iluminación es sub-óptima, los puntos de vistas son elevados y la resolución no es tan alta como se quisiera, sin embargo, aún con todos estos inconvenientes, el detector capta, en algún instante, todas las ocurrencias correctamente ya sea con o sin casco, lo que muestra la robustez del sistema en caso de que en un futuro se desee gatillar una alarma a partir de las detecciones. En general, las grabaciones proporcionadas por PSINet tienen una calidad y resolución mejores que este video, por lo que se espera un rendimiento igual o mejor.

Al comparar con otros detectores aplicados a la misma tarea, 36.82 % es el mAP obtenido para la clase casco en trabajos de investigación por Yange Li et al.[39] de la universidad *South Central University* en China. Esto indica que el rendimiento obtenido de 80 % mAP permite competir con el estado del arte y detectores actuales.

Capítulo 7

Trabajo futuro y Conclusión

7.1. Trabajo futuro

Si bien cada uno de los módulos tiene un correcto funcionamiento, cada uno de estos podría ser mejorado si se desarrollan más ciertas líneas de investigación. Es posible dividir éstas mejoras en dos partes: Entrenamiento de los modelos y heurística de las detecciones.

Respecto al entrenamiento, una considerable mejora sería incrementar tanto la cantidad como la calidad de los datos, es decir, obtener un mayor número de imágenes pero, además, incorporando variaciones como las siguientes: En el detector de máscaras podrían incorporarse muestras de EPP de COVID como escudo facial para poder detectarlas o una nueva clase de mascarilla "mal puesta". También sería útil una mayor cantidad de muestras de personas utilizando mascarilla y lentes ya que se observa en algunos videos que el desempeño disminuye cuando se presentan estos casos; en el detector de casco, sería útil disponer de mayor dataset de chalecos para una mejor generalización, y sobre todo un mayor conjunto de datos de cascos desde un punto de vista elevado, como usualmente se encuentran las cámaras de seguridad, lo que permitiría mejorar la calidad de las detecciones.

En cuanto a la heurística de las detecciones, para el módulo de distanciamiento, un aspecto mejorable sería el hecho de que, actualmente, contabiliza como infracción, aquellos grupos de personas que se encuentran juntos desde el comienzo de la escena, cuando podría de tratarse de personas que viven juntas, o son un grupo permitido a tener cercanía social, para mejorar esto podría evaluarse el estado de compañía de cada persona al entrar a la escena para luego evaluar si este estado cambia, en cuyo caso se incurriría en una infracción. En la heurística del detector de distanciamiento, se podría definir la distancia mínima de manera más inteligente, en vez de preguntarla al usuario, si se obtiene el promedio de las alturas de las personas detectadas, de manera de obtener la escala a la que se observa la gente en la escena, aproximando así la distancia equivalente a los 2 metros de distanciamiento social.

Es evidente que existe una gran cantidad de mejoras y desafíos no resueltos en torno a los módulos desarrollados, por lo que al haber demostrado el correcto funcionamiento de estos, cualquiera podría ser considerado un punto de partida para una posterior línea de investigación.

7.2. Conclusión

Utilizando los más recientes avances en el ámbito de visión computacional, fue posible desarrollar cada uno de los módulos planteados para obtener resultados competitivos respecto del estado del arte. Al combinar estos módulos se puede originar un sistema que combate varios de los dolores que más aquejan hoy el sector minero, haciendo especial énfasis en la mitigación del COVID-19.

Cada uno de los objetivos específicos se cumplió a cabalidad, a pesar de las dificultades afrontadas al trabajar de manera absolutamente remota debido a la pandemia del COVID-19. Esto motivó el uso de nuevas tecnologías, como el uso de poder de cómputo remoto o nuevas dinámicas de trabajo y sincronización con el equipo.

Respecto a las detecciones, los desempeños de los cuatro módulos se encuentran en el rango entre 70 % y 90 %, ubicándolos en una posición competitiva respecto a otras implementaciones de detectores. Estos valores permiten nombrarlos como potenciales herramientas para mejorar la seguridad operacional y prevenir los contagios por COVID-19. Ambas redes utilizadas entregaron resultados positivos a pesar de utilizar técnicas totalmente distintas. En éste ámbito, la interfaz de YOLOv5 hizo el trabajo más amigable en el proceso de entrenamiento. Más aún al trabajar con Google Colab, lo que permitió obtener gran poder de cómputo, esencial para el procesamiento de los datos.

La integración de cada modelo detector con la heurística adecuada permitió generar cuatro programas, enfocados en las necesidades de PSINet, que podrían servir de cimientos para un detector en tiempo real. La existencia de múltiples puntos de operación para cada detector, posibilita que cada programa cumpla con las condiciones de detección para cada tarea asignada.

Las diferencias de rendimiento entre distintos datasets dan cuenta de la vulnerabilidad de los modelos frente a la calidad de los videos a procesar, donde elementos como artefactos o mala iluminación empeoran la calidad de las detecciones. También existen potenciales mejoras para cada uno de los módulos como combatir los falsos negativos al intentar detectar rostros con mascarilla y lentes de sol; la vulnerabilidad del detector de casco frente a distintos colores y formas o la interpretación como infracción, al distanciamiento social, de parejas de personas que entran y salen juntas en una escena.

El desarrollo de estos sistemas representa un primer paso en cuanto a las medidas de prevención de riesgos mediante visión computacional, ya sean frente a accidentes o sanitarias. Si bien ya existen avances en éstas áreas, el trabajo realizado abre posibilidades a una nueva iteración con múltiples mejoras para generar modelos más robustos que sean capaces de incidir efectivamente en el día a día de las personas.

Apéndice

Glosario

- EPP: Elemento de protección personal. Equipo o dispositivo destinado para ser utilizado por el trabajador, para protegerlo de los riesgos y aumentar su seguridad o su salud en el trabajo.
- SERNAGEOMIN: Servicio nacional de geología y minería. Es un organismo técnico, que forma parte de la Administración Central del Estado a través del Ministerio de Minería, responsable de generar y disponer de información de geología básica y de recursos y peligros geológicos del territorio nacional, para el bienestar de la ciudadanía y al servicio de los requerimientos del país, y de regular y fiscalizar el cumplimiento de estándares y normativas en materia de seguridad y propiedad minera, para contribuir al desarrollo de una actividad minera sustentable y socialmente responsable.
- CCTV: Circuito cerrado de televisión. Tecnología de videovigilancia diseñada para supervisar una diversidad de ambientes y actividades. Se denomina *cerrado*, ya que todos sus componentes están enlazados.
- *Pipelines*: Serie de pasos a seguir para obtener un efectivo procesamiento de datos para cumplir un objetivo específico.
- *Bounding box*: Figura geométrica rectangular que contiene un elemento de interés en una imagen.
- *Softmax*: Función que convierte un vector de K valores reales a un vector de K valores reales cuya suma es igual a 1.
- *Feature map*: Representación de una imagen obtenido a partir de un filtro sobre una otra, lo que permite acentuar distintas características.
- *Backbone*: Parte de una arquitectura de red neuronal, encargada de extraer características para codificar el input en una representación en particular.
- *Pooling*: Bloque de una red neuronal dedicado a reducir la dimensionalidad de la red. La técnica más utilizada es *max pooling*, filtro que toma los máximos en coordenadas especificas de la capa previa.
- Benchmark: Punto de referencia o estándar utilizado para comparar modelos.

Bibliografía

[1] EY, "Los 10 principales riesgos de la industria minera," 2019-2020.

[2] S. de Seguridad Social, "Circular 3336," 2018.

[3] R. Energia, "Impacto del coronavirus en la minería chilena," 2020.

[4] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019.

[5] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, yxNONG, A. Hogan, lorenzomammana, AlexWang1900, A. Chaurasia, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Durgesh, F. Ingham, Frederik, Guilhen, A. Colmagro, H. Ye, Jacobsolawetz, J. Poznanski, J. Fang, J. Kim, K. Doan, and L. Yu, "ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration," Jan. 2021.

[6] S. de Seguridad Social, "Vii memoria anual, sistema nacional de seguridad y salud laboral," 2020.

[7] C. C. del Cobre, "Producción cobre de mina por empresa," 2020.

[8] Sernageomin, "Normativa de seguridad minera," 2018.

[9] Sernageomin, "Accidentabilidad minera año 2019," 2020.

[10] d. P. Meritxell Freitas, "Alerta en la industria minera de chile por miles de contagios de sus trabajadores," 2020.

[11] Sernageomin, "Sernageomin desarrolla fuerte presencia de fiscalización de medidas de seguridad y prevención por covid-19 en faenas mineras," 21 junio 2020.

[12] Codelco, "Presencia mundial," 2020.

[13] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," 2019.

[14] K.-C. Song, Y.-H. YAN, W.-H. CHEN, and X. Zhang, "Research and perspective on local binary pattern," *Acta Automatica Sinica*, vol. 39, p. 730–744, 06 2013.

[15] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157 vol.2, 1999.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893 vol. 1, 2005.

[17] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014.

[19] V. Ndonhong, A. Bao, and O. Germain, "Wellbore schematics to structured data using artificial intelligence tools," 04 2019.

[20] P. with Code, "Real-time object detection on coco dataset. leadeboard.," 2020.

[21] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018.

[22] Z. Liu, T. Zheng, G. Xu, Z. Yang, H. Liu, and D. Cai, "Training-time-friendly network for real-time object detection," 2019.

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[26] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," 2018.

[27] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.

[28] B. Russell, A. Torralba, K. Murphy, and W. Freeman, "Labelme: A database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, 05 2008.

[29] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "Mot20: A benchmark for multi object tracking in crowded scenes," *arXiv:2003.09003[cs]*, Mar. 2020. arXiv: 2003.09003.

[30] N. Wojke and A. Bewley, "Deep cosine metric learning for person re-identification," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 748–756, IEEE, 2018.

[31] P. Bhandary, "Open source computer vision library." https://github.com/itseez/opencv, 2020.

[32] A. Rosebrock, "Covid-19: Face mask detector with opencv, keras/tensorflow, and deep learning," 4 mayo 2020.

[33] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild:

A database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

[34] Itseez, "Mask classifier." https://github.com/prajnasb/observations, 2015.

[35] N. D. Nath, A. H. Behzadan, and S. G. Paal, "Deep learning for site safety: Real-time detection of personal protective equipment," *Automation in Construction*, vol. 112, p. 103085, 2020.

[36] J. Li, H. Liu, T. Wang, M. Jiang, S. Wang, K. Li, and X. Zhao, "Safety helmet wearing detection based on image processing and machine learning," pp. 201–205, 02 2017.

[37] S. Gupta, R. Kapil, G. Kanahasabai, S. S. Joshi, and A. S. Joshi, "Sd-measure: A social distancing detector," *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, Sep 2020.

[38] M. Jiang, X. Fan, and H. Yan, "Retinamask: A face mask detector," 2020.

[39] Y. Li, H. Wei, Z. Han, J. Huang, and W. Wang, "Deep Learning-Based Safety Helmet Detection in Engineering Management Based on Convolutional Neural Networks," *Advances in Civil Engineering*, vol. 2020, p. 9703560, Sept. 2020. Publisher: Hindawi.