Hacia una detección precisa de cascos de seguridad en tiempo real a través de un método basado en el aprendizaje profundo

Towards real-time accurate safety helmets detection through a deep learning-based method

Roger Max Calle Quispe[1] https://orcid.org/0009-0004-7343-1319
Maya Aghaei Gavari[2] https://orcid.org/0000-0002-2648-5945
Eduardo Aguilar Torres[1]* https://orcid.org/0000-0002-2463-0301

Recibido 06 de abril de 2023, aceptado 07 de junio de 2023
Received: April 06, 2023 Accepted: June 07, 2023

RESUMEN

La seguridad laboral en la industria es una actividad fundamental debido a la gestión de los controles necesarios que deben estar presentes para mitigar los riesgos laborales y las consecuencias de los accidentes. En estos controles se incluye la verificación del uso de equipamiento de protección personal (EPP), en especial el uso de cascos de seguridad, que tiene vital importancia para reducir consecuencias graves o fatales causados por impactos en la cabeza. Últimamente se han desarrollado investigaciones basadas en el aprendizaje profundo que detectan personas con o sin cascos de seguridad. En estas se ha evidenciado una mejora significativa para el problema de detección de objetos en general y para cascos en particular, por medio de métodos basados en la familia YOLO. En este trabajo, se propone contribuir principalmente en analizar el rendimiento de un novedoso modelo de la familia YOLO que no ha sido evaluado anteriormente en este problema. Específicamente, se evalúa el rendimiento de Scaled-YOLOv4 sobre dos bases de datos públicas, las cuales se seleccionaron luego de una revisión exhaustiva de la literatura sobre conjuntos de datos propuestos para resolver distintos problemas de detección de objetos en el marco de la seguridad laboral. Como resultado se evidencia que Scaled-YOLOv4 logra mejorar el desempeño en términos de mAP y F1-score con respecto a los trabajos previos evaluados en ambas bases de datos. Además, a partir de esta revisión, se genera y se pone a disposición una lista depurada de bases de datos públicas para este propósito.

Palabras clave: Deep Learning, Detección de cascos de seguridad, Scaled-YOLOv4, Bases de datos de cascos de seguridad.

ABSTRACT

Occupational safety is a fundamental activity in industries and revolves around the management of the necessary controls that must be present to mitigate occupational risks. These controls include verifying the use of Personal Protection Equipment (PPE). Within PPE, safety helmets are vital to reducing severe or fatal consequences caused by head injuries. This problem has been addressed recently by various research based on deep learning to detect the usage of safety helmets by the present people in the industrial field.

[1] Universidad Católica del Norte. Departamento de Ingeniería y Sistemas de Computación. Antofagasta, Chile. E-mail: roger.calle01@gmail.com; eduardo.aguilar@ucn.cl
[2] NHL Stenden University of Applied Sciences. Computer Vision and Data Science. Leeuwarden, Países Bajos. E-mail:maya.aghaei.gavari@nhlstenden.com
* Autor de correspondencia: eduardo.aguilar@ucn.cl

These works have achieved promising results for safety helmet detection using object detection methods from the YOLO family. In this work, we propose to analyze the performance of Scaled-YOLOv4, a novel model of the YOLO family that has yet to be previously studied for this problem. The performance of the Scaled-YOLOv4 is evaluated on two public databases, carefully selected among the previously proposed datasets for the occupational safety framework. We demonstrate the superiority of Scaled-YOLOv4 in terms of mAP and F1-score concerning the previous works for both databases. Further, we summarize the currently available datasets for safety helmet detection purposes and discuss their suitability.

Keywords: Deep Learning, Safety helmet detection, Scaled-YOLOv4, Safety helmet datasets.

INTRODUCCIÓN

Los comportamientos inseguros que pueden presentar los trabajadores, las actividades de alto riesgo y la ejecución de trabajos en lugares peligrosos conllevan a menudo a accidentes laborales [1]. Para reducir los riesgos que provocan accidentes, los especialistas de seguridad, tienen el rol de monitorear e identificar los riesgos sobre los controles críticos, condiciones de terreno, estructuras o comportamientos de los trabajadores, para así establecer estándares de seguridad en la organización. Sin embargo, aun así los trabajadores están sujetos a peligros de seguridad en entornos críticos [2]. En este sentido, es necesario velar por el cumplimiento de los protocolos de seguridad para disminuir consecuencias graves derivadas de algún accidente. En concreto, dentro de los protocolos de seguridad, está el uso apropiado de equipamiento de protección personal (EPP), el cual depende de la actividad encomendada. Particularmente, el casco de seguridad es un EPP utilizado para proteger la parte superior de la cabeza del trabajador, siendo una barrera protectora que absorbe las energías de los impactos [3]. Su uso toma vital importancia considerando lo común que resultan las lesiones de la cabeza en países industrializados. Específicamente, este tipo de lesiones fluctúa entre el 3% y 6% de todos los accidentes con consecuencias laborales que suelen ser graves o pueden llegar a ser fatales [3]. Además, considerando que uno de los factores sobre las causas básicas de los accidentes son las circunstancias que dependen del comportamiento de los trabajadores [4], es relevante que las organizaciones incrementen sus capacidades de monitoreo y control de conductas riesgosas, como el no uso de los cascos de seguridad, con el propósito de mitigar la gravedad de un accidente. Por lo que resulta importante disponer de herramientas adicionales que faciliten el análisis del cumplimiento de los protocolos de seguridad.

Recientemente, varios trabajos relacionados con tecnologías basadas en aprendizaje profundo han sido aplicados con éxito para contribuir en la seguridad laboral en industrias de la construcción y energía [4, 5]. En estos se han considerado principalmente la detección de trabajadores [6], de maquinarias de construcción [7] y de EPP [1, 5, 8-10] con el fin de recopilar información útil para reducir los riesgos de que se produzca un accidente, así como las consecuencias de los mismos. Particularmente, para la detección de EPP, algoritmos de detección de objeto de propósito general como *Single Shot Detection* (SSD) [10], *Faster Region Convolutional Neural Network* (Faster R-CNN) [9] y *You Only Look Once* (YOLO) [1, 5, 8] son adoptados como base de los trabajos propuestos, siendo los métodos basados en YOLO los que han presentado mejor desempeño en términos de *mean Average Precision* (mAP) y *Frame Per Second* (FPS). Tenga en cuenta que diversos EPP han sido objeto de estudio en el estado del arte: la detección de casco y chaleco de seguridad se evaluó en [2, 8, 11]; de casco, botas y guantes en [5]; cascos de diferentes colores en [1]; por nombrar solo algunos. Más relacionado a la investigación propuesta, se encuentran los trabajos que se enfocan en la detección del uso del casco de seguridad por parte de los trabajadores desde una imagen [12, 13]. Generalmente, estos se evaluaron sobre una única base de datos pública, y otras de acceso privado, mediante modelos de aprendizaje profundo basados en diferentes versiones de YOLO [14-19]. Los resultados obtenidos son notables para esta problemática, evidenciando en muchos de ellos un desempeño en tiempo real (> 30 FPS) por encima de un 90% en términos de mAP.

En este trabajo de investigación se analiza el uso de Scaled-YOLOv4 [17] para la detección de cascos de seguridad, método de aprendizaje profundo perteneciente a la familia de YOLO seleccionado

para el estudio debido a los prometedores resultados que ha mostrado en la detección de objetos en general. La evaluación del rendimiento de este método se realiza en dos bases de datos públicas con el fin de analizar el rendimiento del Scaled-YOLOv4 sobre imágenes adquiridas en distintas condiciones y también para facilitar la replicación de los resultados obtenidos y la comparativa de rendimiento para los trabajos posteriores. Las contribuciones que se derivan del presente trabajo principalmente son: 1) Se analiza por primera vez el uso de Scaled-YOLOv4 para la detección de cascos de seguridad, entre todos los métodos disponibles para la detección de objetos; 2) Se presentan los beneficios de los métodos utilizados para la detección del casco de seguridad sobre dos bases de datos públicas, en que se resaltan los resultados obtenidos con Scaled-YOLOv4 logrando un desempeño comparable e incluso mejores a los evidenciados en el estado del arte; y 3) Se provee de una lista de bases de datos disponibles para la detección de objetos en el contexto de la seguridad laboral con el fin de facilitar a los investigadores su elección dependiendo de la problemática que se desea abordar.

En la siguiente sección se resumen los trabajos relacionados. Posteriormente, se describen los materiales y métodos usados en la experimentación. Luego se detalla la configuración experimental. Después se presentan y discuten los resultados obtenidos. Por último, se termina con las conclusiones más relevantes derivadas de esta investigación y el trabajo futuro.

TRABAJOS RELACIONADOS

La detección de objetos es una tecnología novedosa para la aplicación en aspectos de seguridad en la industria. Existe un gran interés en la comunidad científica por el desarrollo de métodos de detección de EPP y particularmente los cascos de seguridad basados en algoritmos de aprendizaje profundo. A continuación, se describen algunos de los trabajos que más se relacionan con esta investigación.

Diversos EPP han sido de interés de la comunidad científica para su detección a partir de imágenes. En [1], se evalúa la detección mediante distintos métodos de la familia YOLO sobre una base de datos que contempla 6 clases de objetos: la persona, el chaleco y el casco que puede estar presente en cuatro colores diferentes dependiendo del rol del trabajador. EPP de diferentes colores también se analizaron en [11], quiénes evaluaron la capacidad de diversos métodos de aprendizaje automático para diferenciar los chalecos de seguridad de colores fluorescentes con respecto a los colores presentes en otra vestimenta que pueden usar los trabajadores. Otros EPP como los guantes y botas, además del casco, se detectaron en [5]. En este artículo se desarrolló un método de nombre *wear-enhanced* YOLOv3 (WEYOLOv3) que resalta los detalles de los operadores a partir de un preprocesamiento de imágenes mediante una corrección gamma con el fin de ajustar el contraste y con ello resolver el problema de luz de fondo presente en los datos. También aplican el algoritmo k-means++ [20] para agilizar y obtener una detección más adecuada de las cajas delimitadoras. Además, en [21], se realiza la detección de varios EPP ubicados en la cabeza, tales como: cascos, gafas de seguridad, mascarillas quirúrgicas, entre otros. El método usado es YOLOv5 [19] y se evaluó sobre una base de datos construida a partir de dos bases de datos públicas (PictorPPE [8] y HHW [12]) y, una colección de imágenes propia descargada de la web. Un análisis más en profundidad se realizó en [8], quienes no solo detectan la persona, casco y chaleco de seguridad por separado, sino que también proporcionan diversas estrategias basadas en el método YOLOv3 [16] para determinar si el trabajador está utilizando el chaleco de seguridad, el casco o ambos.

Más relacionado con nuestra investigación, para la detección binaria de cascos de seguridad (trabajadores con o sin casco), existen principalmente dos bases de datos públicas para este propósito: HHW [12] y SHWD [22]. En cuanto a HHW, los mismos autores propusieron un método basado en SSD [23] y MobileNet [24] para este propósito, el cual consta de tres principales componentes: 1) el primero es el *backbone* para extraer mapas de características de escalas múltiples; 2) el segundo, es un módulo de arriba-abajo para combinar las características de las capas superiores e inferiores, entregando una mejora en la precisión para la detección de objetos pequeños; 3) por último, el tercero, es un bloque residual para realizar la predicción, que tiene como entrada el mapa de característica desde las características piramidales. Esta misma base de datos ha sido utilizada para evaluar métodos

que se enfocan en resolver la detección de cascos de seguridad enfocados a dispositivos embebidos [13, 25]. En [13], se analizó la detección de casco de seguridad en tiempo real mediante un sistema inteligente local. El sistema inteligente se compone por una unidad de procesamiento de visión (VPN por sus siglas en inglés) y un sistema embebido que incluye una RaspberryPI 4, una *Intel RealSense Depth Camera D435* y un *Intel Neural Compute Stick 2*. Para la detección de objetos se utilizó YOLOv4 [18] y YOLOv4-tiny, logrando un tiempo de respuesta de 0,8 FPS y 4,7 FPS respectivamente. En el caso de YOLOv4 se evidenció un mejor desempeño principalmente sobre todo en la detección de objetos pequeños. En cuanto a [25], se evalúa el rendimiento de diversos modelos ocupando computadoras de placa única multifuncional. Los modelos se entrenaron sin la necesidad de una GPU, en su lugar se ocupó un *Edge TPU Acceletator* conectado a una RaspberryPI 4 y la librería TensorFlow Lite. En la discusión de resultados se realizó un análisis comparativo de cada uno de los modelos evaluados con respecto a las medidas de evaluación en términos de FPS y mAP y se concluye que YOLOv5-s es aquel que logró el mejor desempeño.

Con respecto a SHWD, varios trabajos han sido publicados recientemente [26-33]. La mayoría desarrollan nuevas propuestas basadas en alguna de las versiones disponibles del detector YOLO. En YOLOv3 se han logrado mejoras en el desempeño aplicando un preprocesado de datos [28], incorporando ajustes en el diseño de la red [32] y seleccionado funciones de pérdidas más apropiadas [32]. Específicamente, se reconstruyen imágenes en alta resolución a partir de una imagen de baja resolución mediante la aplicación del algoritmo de reconstrucción de superresolución (ESPCN [34, 35]) en [28] y se incorporan mecanismos de atención [36] al modelo y también se aumenta la precisión cuando se utilizan las funciones de pérdidas GIOU-loss [37] y CIoU-loss [38] en [32]. Mecanismos de atención también han sido incorporados en YOLOv4 [27], donde se propone usar un *Pyramid Split Attetion* (PSA) [39] para realizar el proceso de escala múltiple, permitiendo resolver los problemas de detección de objetos pequeños y con baja precisión. También YOLOv4 ha sido optimizado en [29] mediante el reemplazo de las capas convolucionales apiladas en el *neck* (entre el *backbone* y el *head* de la red) por estructuras semejantes a *Cross-Stage Partial Network* (CSPNet) [40], y con ello son capaces de reducir la memoria requerida y mejorar el rendimiento. Por último, métodos basados en YOLOv5 para la detección de casco de seguridad han sido propuestos recientemente [26, 30, 31, 33]. Las mejoras realizadas en la literatura revisada son: realización de aumento de datos mediante de CutMix [33, 41]; adaptación de función de pérdida considerando un nuevo criterio para selección de muestras positivas [26]; integración de un filtrado basado en densidad con *Non-Maximum Suppression* (NMS) [8, 26]; y ajuste de la red mediante la incorporación de un mecanismo de atención basado en *Squeeze-and-Excitation Networks* (SENet) [30, 40, 42].

Por último, Sadiq y otros [31], realizaron un análisis comparativo de diversos métodos de detección sobre múltiples bases de datos públicas, dentro de ellas HHW y SHWD. Los autores formularon 3 subproblemas de detección de objetos y realizaron la comparativa entre los modelos YOLO [16, 18], SSD [23] y Faster R-CNN [43]. Para ello, el estudio abordó subproblemas con los siguientes objetivos: 1) detectar únicamente los cascos de seguridad; 2) detectar el casco y ausencia de este (solo sobre SHWD); y 3) detectar múltiples objetos que se podrían usar en la cabeza (ej. casco, gorro, turbante, entre otros). Se muestra que los mejores resultados en términos de tiempo de respuesta y mAP se obtuvieron por el modelo YOLOv5-m.

Como se puede apreciar en los trabajos relacionados, aunque existe una gran diversidad de métodos propuestos para la detección de objetos en el marco de la seguridad laboral, como es la detección de distintos tipos de EPP [1, 5, 8, 11, 21] y, en especial la detección del uso o no de los cascos de seguridad [12, 13, 25, 26-33], la mayoría de estos métodos [12, 13, 21, 25, 26, 27, 29-33] han sido evaluados sobre una única base de datos pública para el problema de detección de detección binaria de cascos. Además, tenga en cuenta que en muchos se realizan ajustes en los datos proporcionados por las bases de datos públicas, modificando la cantidad de instancias y corrigiendo las anotaciones incorrectas [25, 26, 29]. Esto último, complejiza poder evaluar comparativamente los desempeños logrados por otros métodos sobre los mismos datos. Por otro lado, en términos generales, se destaca que los mejores resultados basados en las métricas de mAP y tiempo de respuesta han sido obtenidos por medio de

métodos basados en la familia YOLO [9-11, 26-33]. En el trabajo propuesto se continúa profundizando en el análisis de desempeño de algoritmos de la familia YOLO, en este caso se analiza por primera vez el método Scaled-YOLOv4 para la detección de objetos en ámbitos de seguridad. Además, para facilitar la comparativa de desempeño se evalúa el rendimiento de este método sobre dos bases de datos públicas usando los datos originales que se disponen en ambas.

MATERIALES Y MÉTODOS

En esta sección se describe la evolución del algoritmo de detector de objetos YOLO desde la primera versión hasta la versión Scaled-YOLOv4, que es utilizada en la experimentación, y las bases de datos públicas seleccionadas para la evaluación del método seleccionado.

Detección de Objetos mediante Scaled-YOLOv4

YOLO [14] se propuso con la finalidad de proveer detecciones de objetos precisos en tiempo real. A diferencia de los métodos predecesores, YOLO es un detector de objetos de una única etapa, es decir, permite realizar la detección de los cuadros y la clasificación de estos directamente sin la necesidad de requerir previamente una la generación de regiones candidatas. Su arquitectura se compone por 24 capas convolucionales seguidas de 2 capas que completamente conectadas, produciendo como resultado un tensor de 7 x 7 x (B x 5 + C), donde el número 5 representa a las 5 predicciones que se obtienen para cada *bounding box*, estas son: coordenadas x, y del centro; altura y ancho; y confianza asociada a la predicción del objeto. Además, B corresponde a la cantidad de *bounding boxes* que se detectan en cada de las ubicaciones de una cuadrícula de 7 x 7 ubicada sobre la imagen y C a la cantidad de clases. Posteriormente, el método YOLO ha ido escalando en distintas versiones, como lo es la versión de YOLO9000 [15], donde mejora más la precisión, rapidez y su robustez para la detección de instancias de miles de objetos, mediante la incorporación de los siguientes elementos: *Batch Normalization*, para estabilización del entrenamiento; *High Resolution Classifier*, en que se ajusta el tamaño de entrada del clasificador de 224 x 224 a 448 x 448; *Anchor Boxes*, para la predicción de los *bounding boxes* permitiendo con este ajuste eliminar el uso de capas completamente conectadas; *Fine-Grained Features*, se realizan las predicciones sobre un mapa de características de mayor tamaño (13 x 13); *Multi-Scale Training*, entrenamiento utilizando imágenes con múltiples resoluciones, las cuales son ajustadas en tiempo de entrenamiento aprovechando que el modelo es completamente convolucional y por lo tanto no depende de un tamaño entrada fijo; entre otras mejoras. El modelo resultante permitió incrementar el rendimiento por más de 15% con respecto a la primera versión. La tercera versión, YOLOv3 [16], amplía la profundidad del clasificador de la versión anterior, Darknet19, desde 19 a 53 capas convolucionales para realizar la extracción de características mediante el nuevo clasificador de nombre Darknet53. También produce una mayor cantidad de *bounding boxes*, los cuales son obtenidos a partir de mapas de características de 3 diferentes escalas: 13 x 13, 26 x 26 y 52 x 52. Además, reemplaza la función *softmax* para la predicción de las clases por la función sigmoide. La cuarta versión, YOLOv4 [18], mejora el clasificador Darknet53 por medio de la técnica *Cross-Stage Partial* (CSP), produciendo el clasificador CSPDarknet53 que mejora tanto el tiempo de respuesta como la precisión y permite además reducir la cantidad de cómputo del clasificador original. Además, incrementa el campo receptivo por medio de un bloque de *Spatial Pyramid Pooling* (SPP) [44] e incorpora PANet [45] para agregar las características extraídas a diferentes niveles de la CNN. Después está YOLOv5, versión que fue liberada tempranamente y aún se encuentra en proceso de desarrollo, y por lo tanto no tiene ningún artículo científico asociado. Finalmente, la última versión publicada es Scaled-YOLOv4, construida usando el mismo *framework* que YOLOv5, la cual rediseña la arquitectura YOLOv4 con el fin de proveer de un diseño eficiente en términos de precisión y velocidad para diferentes tamaños de entrada.

Scaled-YOLOv4

Scaled-YOLOv4 es un detector de objetos diseñado con la intención de escalar eficientemente la arquitectura de la red teniendo en cuenta el tamaño de entrada de las imágenes. La base del diseño de este modelo es equivalente a YOLOv4, sin embargo, se consideran varias optimizaciones que permiten mejorar los recursos computacionales requeridos, así como un mejor equilibrio entre precisión y tiempo de respuesta. Entre las mejoras se destaca la modificación realizada a CSPDarknet53, en el que se

incorporan capas residuales; se aplica la técnica CSP sobre la arquitectura PAN (CSPPAN) y se agrega un módulo SPP en la posición media logrando reducir la cantidad de parámetros del modelo y una mejor precisión; por último, en la parte superior de la red se mantiene el diseño propuesto por YOLOv3, es decir se realiza las detecciones sobre mapas de características de distintas dimensiones. Más detalles en cuanto a la arquitectura Scaled-YOLOv4 se puede ver en la Figura 1. Tenga en cuenta que los autores generaron sistemáticamente modelos pequeños y grandes llamados Scaled-YOLOv4-tiny y Scaled-YOLOv4-large, respectivamente. ScaledYOLOv4-tiny se optimizó teniendo en cuenta la cantidad de cómputo, el tamaño del modelo y la limitación de los recursos de hardware periférico, para ser utilizado en dispositivos de gama baja. En cuanto a Scaled-YOLOv4-large se diseñó con el propósito de alcanzar una alta precisión y se sugiere su uso por medio de una GPU en la nube. En ambos casos se demuestra que pueden ser ejecutados en tiempo real. Por último, es interesante destacar que las distintas versiones de Scaled-YOLOv4-large (P5, P6 y P7) otorgan resultados igual de precisos que las distintas versiones de EfficientDet (D5, D6 y D7) [46], sin embargo, proporcionan una velocidad de inferencia de alrededor de 3 veces superior.

Bases de datos de imágenes con anotaciones de EPP

Múltiples bases de datos han sido construidas con imágenes de trabajadores en entornos de la industria de la construcción y sus respectivas anotaciones en cuanto a los EPP que aparecen en ellas. Sin embargo, solo unas pocas se encuentran inmediatamente disponibles mediante un enlace público, sin la necesidad de solicitar autorización a sus propietarios para su acceso. En la Tabla 1, se muestran las distintas bases de datos disponibles para la detección de objetos en ámbitos de seguridad en las industrias, donde se ordena por año de publicación y se visualiza las distintas características como, la cantidad de imágenes e instancias, las anotaciones y la compatibilidad de las anotaciones para realizar la detección de casco y no casco. De todas las bases de datos analizadas, encontramos dos bases de datos que cuentan con anotaciones apropiadas para el problema en cuestión, estas son: 1) HHW [12] y 2) SHWD [22].

Hard Hat Workers (HHW)

HHW [12] es una base de datos que contiene un total de 7.064 imágenes distribuidas en 5.298 para el entrenamiento y las restantes 1.766 para propósitos de validación. Todas las imágenes se encuentran manualmente etiquetadas en formato Pascal VOC, en las que se indican la localización y clase correspondiente a las instancias de los objetos que aparecen en la imagen. En total se disponen de alrededor de 26.633 instancias, de las cuales 19.852 corresponden a la clase *helmet* y el resto se distribuye en 3 clases: *head*, *person*, *others*. Una muestra de las imágenes disponibles en esta base de datos y sus anotaciones se pueden observar en

Fuente: Scaled-YOLOv4: Scaling Cross Stage Partial Network [17].

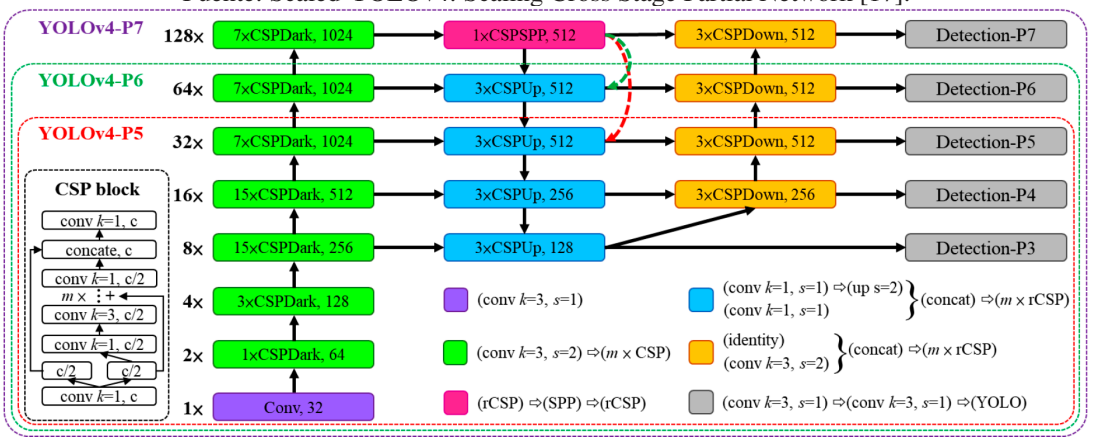

Figura 1. Arquitectura de la red neuronal Scaled-YOLOv4 en la que se ilustra cómo es el escalado entre las diferentes versiones que ofrece (P5, P6 y P7).

Tabla 1. Bases de datos públicas para la detección de objetos en el marco de la seguridad laboral.

Nombre	Año de Publicación	Total imágenes	Cantidad de instancias	Anotaciones	Etiquetas Casco/ no-Casco
Hard Hat Workers (HHW) [12]	2019	7.064	26.633	Helmet - Head	Si
Safety Helmet Wearing Dataset (SHWD) [22]	2019	7.581	120.558	Hat - Person	Si
Pictor-V3 (CROWD-SOURCED) [8]	2020	774	2.496	Hat - Vest - Worker	Parcial
Alberta Construction Image Dataset (ACID) [47]	2021	10.000	15.767	excavator, compactor, dozer, grader, dump truck, concrete mixer truck, wheel loader, backhoe loader, tower crane, mobile crane	No
CHV Dataset [1]	2021	1.330	9.209	Person - Vest - Blue - Red - White - Yellow	Parcial

la Figura 2. En los experimentos se considera la misma distribución de los datos propuesta por los autores y realizamos el entrenamiento para detectar las instancias de los objetos etiquetados como *helmet* (casco) y *head* (no-casco).

Safety Helmet Wearing Dataset (SHWD)
SHWD [22] es una base de datos con imágenes etiquetadas para realizar la detección de EPP, específicamente con respecto al casco de seguridad (*hat*) o la cabeza de la persona (*person*). En total consiste en 7.581 imágenes que incluyen las anotaciones de 9.044 instancias del objeto *hat* (casco) y 111.514 del objeto *person* (no-casco). Las imágenes se obtuvieron desde dos fuentes, las que contienen cascos de seguridad desde la web mediante motores de búsqueda y las restantes desde la base de datos SCUT-HEAD [48]. Luego de recopilar todas las imágenes, se revisaron para minimizar los errores en las anotaciones. Finalmente, con las imágenes corregidas, se generaron anotaciones en formato Pascal VOC. Para propósitos experimentales, esta base de datos dispone de 5.457 imágenes de entrenamiento, 607 de validación y 1.517 de prueba. Con la intención de obtener resultados comparables con los trabajos previos, se decide utilizar 6.064 imágenes para el entrenamiento (unión de conjuntos de entrenamiento y validación) y las 1.517 para la

Figura 2. Imágenes de ejemplo para la base de datos HHW. El recuadro de color rojo representa la clase no-casco, el azul la clase casco y el verde la clase *person.*

validación y prueba. En la Figura 3 se muestra una imagen representativa de esta base de datos junto con las anotaciones que proporciona.

CONFIGURACIÓN EXPERIMENTAL

El método de detección de objetos Scaled-YOLOv4 se seleccionó debido al notable desempeño que se observó en la revisión de la literatura sobre bases de datos de propósito general, permitiendo además que se realice la detección en tiempo real. En este método se ofrecen cinco variantes (*tiny*, CSP, P5, P6 y P7), en que se adecua la red neuronal según el tamaño de la imagen de entrada. De todos modos, estas variantes pueden usarse para tamaños distintos al considerado para su diseño. Teniendo en cuenta el rendimiento ofrecido por las distintas variantes sobre la detección de objetos en general y también la posibilidad de ejecución en tiempo real, se decide evaluar el desempeño que ofrece la variante P5 de Scaled-YOLOv4 (Scaled-YOLOv4-p5) para la detección binaria de cascos de seguridad. Para la experimentación utilizamos 896 X píxeles para las imágenes de entrada y de únicamente modificamos la última capa de Scaled-YOLOv4-p5 considerando la cantidad de objetos distintos que se desean detectar. En este caso se consideran 2 clases en la última capa.

Previo al entrenamiento, para cada base de datos objetivo se calculan los *anchors*, 12 en total, con el propósito de usar dimensiones de referencia para los *bounding boxes* más acordes a las proporciones de los objetos que se desean detectar. El cálculo se realiza por medio de un algoritmo genético que usa los centroides dados por k-means [20] como condición inicial y una función de idoneidad que considera el mejor *recall* al comparar el ancho y alto de los *anchors* con respecto a los *bounding boxes* del conjunto de entrenamiento. Una vez ajustado los *anchors*, se inicializan los pesos con los valores obtenidos del mismo modelo pre entrenado sobre la base de datos COCO [49]. Posteriormente, se inicia el entrenamiento durante 300 épocas considerando un subconjunto de 4 imágenes para cada iteración optimizando la función de pérdida por medio de *Stochastic Gradient Descent* (SGD) con *Nesterov momentum* de 0,937 y *weight decay* de 5e-4. Con respecto a la función de pérdida (L), está se basa en tres componentes: 1) *Regression Loss* (*RL*): pérdida asociada a los *bounding boxes* predichos, 2) *Classification Loss* (*CL*): pérdida asociada a la clasificación dada con respecto a la instancia contenida en cada *bounding box* y 3) *Objectness Loss* (*OL*): pérdida asociada a la existencia de un objeto en cada *bounding box*. Para *RL* se utiliza la pérdida *Complete Intersection over Union* (CIoU), que corresponde a una agregación con respecto a el área sobrepuesta, la distancia entre los centros y la diferencia de proporción de pares de bounding boxes (el real y el predicho). En el caso de *CL* y

Figura 3. Imágenes de ejemplo para la base de datos SHWD. El recuadro de color rojo representa la clase no-casco y el azul la clase casco.

OL, la pérdida *Binary Cross Entropy* (BCE) es utilizada. El cálculo de la pérdida total se muestra en la ecuación (1):

$$L = 0{,}05 \times RL + 0{,}5 \times CL + 1{,}0 \times OL \quad (1)$$

En cuanto al esquema de aprendizaje, las primeras 3 épocas se utilizan de calentamiento. Para ello, se ocupa una tasa de aprendizaje de 1e-5 que incrementa linealmente al final de cada iteración hasta alcanzar 1e-2. Luego cuando el calentamiento concluye, la tasa de aprendizaje disminuye gradualmente al final de cada época de entrenamiento mediante un decaimiento cosenoidal.

Con respecto al preprocesamiento de datos, las imágenes y los *bounding boxes* son normalizados en una escala de 0-1. Además, en tiempo de entrenamiento, para evitar que el modelo se sobreajuste se aplican varias técnicas de aumento de datos para la detección de objetos, tales como: variación aleatoria en el espacio de color HSV, desplazamientos horizontales y verticales, recortes sobre la imagen original, escalado, volteos horizontales y Mixup [50]. Tenga en cuenta que en la etapa de prueba se utiliza la imagen original redimensionada sin aplicar ninguna estrategia adicional de aumento de datos.

Por otro lado, para la evaluación cuantitativa de los resultados se usa como medida de desempeño el mAP@.5 y mAP@.5:.95. Es decir, se calcula el *Average Precision* (*AP*) para cada clase individualmente y luego son promediados. Tenga en cuenta que @.5 y @.5:.95 denotan el umbral utilizado para que una detección sea considerada correcta. En el caso de mAP@.5 la IoU debe ser superior al 50%. Con respecto a @.5:.95, se evalúa la detección usando umbrales que van desde el 50% al 95% con incrementos de 5%, y se considera el promedio de los mAP para cada umbral individual como resultado final. A continuación, se definen los conceptos *True Positive* (*TP*), *False Positive* (*FP*) y *False Negative* (*FN*) en el contexto de la detección de objetos para comprender de mejor manera cómo se calcula el *AP*. *TP* se refiere a las detecciones que satisfacen el umbral de IoU predefinido y además se identifica correctamente la clase del objeto; *FP* a las detecciones que se encuentran duplicadas o que no satisfacen el umbral de IoU; y *FN* a las detecciones que satisfacen el umbral pero la clase predicha es incorrecta o cuando no se detecta la instancia del objeto en la imagen. Con esto, se obtiene la *Precision* (*P*) y *Recall* (*R*) en la ecuación (2) y ecuación (3), respectivamente.

$$P = \frac{TP}{TP + FP} \quad (2)$$

$$R = \frac{TP}{TP + FN} \quad (3)$$

Luego cada uno de los bounding boxes predichos son ordenados según el nivel de confianza asignada a cada una de esas predicciones para construir la curva Precision-Recall (PR). El último paso consiste en calcular el área bajo la PR curva para determinar el *AP*, comúnmente promediando la *P* obtenida en 11 valores del *R* que van desde 0 a 1 con incrementos de 0,1. Formalmentes el *APAP* se define en la ecuación (4), como sigue:

$$AP = \frac{1}{11} \sum r \in \{0{,}0{,}1 \ldots, 0{,}9{,}1\} p(r) \quad (4)$$

donde *p(r)* es una función que devuelve la *P* obtenida cuando el *R* toma el valor *r*.

En soluciones reales, es relevante tener conocimiento de la capacidad de los modelos para detectar todas las instancias presentes y para evitar instancias erróneas o duplicadas. Por esta razón, además de la medida tradicional usada para la detección de objetos, mAP, se considera la medida F_1. Con esta medida, es posible interpretar estos dos aspectos basados en el puntaje calculado que refleja el balance de los resultados que provee el modelo con respecto a las medidas *P* y *R*. Formalmente F_1 se define en la ecuación (5):

$$F_1 = 2 \times \frac{P \times R}{P + R} \quad (5)$$

Durante la prueba se evaluaron distintas configuraciones para la aplicación de NMS sobre la salida del modelo, en el que se encontró que un valor mínimo de confianza en las predicciones de 0,3 y un umbral de IoU de 0,5 corresponde al ajuste de estos parámetros de posprocesamiento que proporcionan los mejores resultados con respecto a las medidas de mAP@.5, mAP@.5:.95 y F_1.

Todos los experimentos se realizaron en el *framework* de *deep learning* PyTorch v1.10, los cuales fueron ejecutados en un servidor que posee una tarjeta gráfica NVIDIA de gama media (RTX 3060, 12GB de VRAM), 16GB RAM y un procesador AMD Ryzen 5600X.

RESULTADOS

Los resultados experimentales son presentados y discutidos cuantitativamente por medio de las métricas tradicionales utilizadas en la detección de objetos y cualitativamente a través del análisis visual de las detecciones logradas tanto con los casos favorables como desfavorables del método propuesto.

En las Tabla 2 y Tabla 3 se presentan los resultados obtenidos en la detección binaria de cascos de seguridad para las bases de datos SHWD y HHW, respectivamente. También se muestran los desempeños obtenidos por otros métodos propuestos anteriormente. Aunque existen más trabajos que evaluaron sus propuestas sobre estas bases de datos, únicamente se consideran aquellos que usan los mismos datos para el entrenamiento y validación. Las métricas utilizadas para la comparativa son mAP@.5, mAP@.5:95 y F_1. Ésta última es sumamente relevante debido a que nos permite observar cómo es la compensación entre la cantidad de instancias generadas con respecto a las que se deben detectar. Tenga en cuenta que es posible obtener un mAP más alto usando una confianza más baja para el NMS, lo que puede afectar positivamente el *R* (se reconocen más instancias reales) pero negativamente la *P* (se generan detecciones duplicadas) en algunas muestras. Si la compensación entre *P* y *R* no mejora, aunque se logre un mayor mAP, bajar la confianza resultaría contraproducente considerando el uso del método en situaciones reales. En la Tabla 2, se puede observar que el modelo propuesto provee resultados comparables en términos de mAP@.5 y F_1 con respecto a YOLOv5-m. Sin embargo, provee una mejora significativa en términos de mAP@.5:.95, lo cual tiene directa relación con la calidad de las regiones detectadas. Por otro lado, cuando observamos los resultados en ambos objetos separados con respecto a mAP@.5:95, podemos notar que Scaled-YOLOv4-p5 tiene una diferencia de más de 20% entre las instancias de casco y no-casco. La causa de esto puede ser debido a que existe una gran cantidad de objetos pequeños para la clase no-casco y la mayoría de estos son cubiertos principalmente por un único *anchor* (Figura 4). Como consecuencia, se eleva complejidad del modelo para poder comprender las proporciones de todas las instancias pequeñas de la clase no-casco. En cuanto a la Tabla 3, podemos observar que el método propuesto es muy superior al resto, el cual no solo provee un mejor desempeño,

Tabla 2. Resultados en la detección de casco/no-casco para la base de datos SHWD.

Modelo	**Resolución**	**mAP@.5**	**mAP@.5 Casco**	**mAP@.5 no-Casco**	**mAP @.5:.95**	**mAP @.5:.95 Casco**	**mAP @.5:.95 no-Casco**	F_1
ESPCN + YOLOv3 [28]	416 x 416	76,14%	77,6%	74,7%	–	–	–	–
YOLOv4-P [29]	416 x 416	92,77%	93,28%	92,26%	–	–	–	–
YOLOv5 m [31]	640 x 640	94,7%	–	–	61,1%	–	–	93,6%
Scaled-YOLOv4-p5 [17]	896 x 896	94,9%	94,4%	95,4%	63,6%	75,5%	51,8%	93,8%

Tabla 3. Resultados en la detección de casco/no-casco para la base de datos HHW.

Modelo	**Resolución**	**mAP@.5**	**mAP@.5 Casco**	**mAP@.5 no-Casco**	**mAP @.5:.95**	**mAP @.5:.95 Casco**	**mAP< @.5:.95 no-Casco**	F_1
SSD [12]	300 x 300	87,5%	88,7%	86,3%	–	–	–	–
SSD-MobileNet [12]	300 x 300	88,4%	89,4%	87,4%	–	–	–	–
YOLOv4 [12]	416 x 416	94,0%	96,1%	92,0%	–	–	–	–
Scaled-YOLOv4-p5 [17]	440 x 640	96,7%	97,2%	96,3%	67,9%	68,2%	67,7%	95,0%

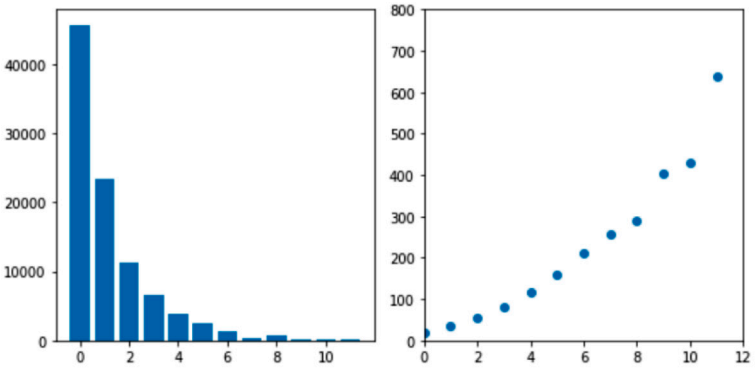

Figura 4. Gráficos que muestran la cantidad de instancias (izquierda) y la raíz cuadrada de las proporciones de los *anchors* (derecha) asociadas a cada *cluster* obtenido sobre el conjunto de entrenamiento de SHWD.

sino que también un resultado balanceado entre las instancias de los distintos tipos de objetos. En este caso, los resultados obtenidos son utilizando una resolución de la imagen más baja de la utilizada originalmente para el diseño de la arquitectura de la red lo que permite mejorar el tiempo de respuesta del mismo sin afectar los resultados.

En la Tabla 4, se evalúa el desempeño de Scaled-YOLOv4-p5 cuando se modifica la resolución utilizada para el entrenamiento. Para este estudió se seleccionan 3 resoluciones distintas, dos que corresponden a aquellas utilizadas generalmente por otros algoritmos (416 x 416 y 640 x 640) y la última a la resolución original (896 x 896). En esta tabla, se usan columnas similares a las de la Tabla 2 y Tabla 3, sin embargo, se incorpora una columna adicional para indicar el tiempo de ejecución (en FPS) que se logra en las distintas pruebas realizadas. Sobre SHWD, se observa que a medida que aumentamos la resolución se logran mejores resultados, en mayor medida cuando se compara el modelo entrenado con imágenes de 416 x 416 en comparación con el modelo entrenado con imágenes de 640 x 640. La ganancia se obtiene principalmente para las instancias de la clase no-casco, las cuales generalmente son más pequeñas. En el caso de HHW, también se observa una mejora cuando se aumenta la resolución la cual se produce desde416 x 416 a 640 x 640 y se estabiliza desde 640 x 640 a 896 x 896. Este efecto se puede deber a que la base de datos HHW cuenta con instancias mejor distribuidas en términos de proporciones, en que el área de la mayoría de estas es de100^2 o superior (Figura 5) y por ende no requieren el hecho de utilizar imágenes de mayores resoluciones no contribuye a la identificación de nuevas instancias (ej. instancias pequeñas no detectadas). Considerando

Tabla 4. Resultados en la detección de casco/no-casco al utilizar el modelo Scaled-YOLOv4-p5 entrenado con imágenes de distintas resoluciones.

Base de datos	**Resolución**	**mAP@.5**	**mAP@.5 Casco**	**mAP@.5 no-Casco**	**mAP @.5:.95**	**mAP @.5:.95 Casco**	**mAP @.5:.95 no-Casco**	F_1	**Tiempo de ejecución (FPS)**
SHWD	416 x 416	86,5%	91,0%	82,1%	57,4%	71,9%	43,0%	89,0%	56
	640 x 640	93,8%	93,3%	94,3%	62,8%	75,0%	50,5%	92,8%	55
	896 x 896	94,9%	94,4%	95,4%	63,6%	75,5%	51,8%	93,8%	35
HHW	416 x 416	96,3%	97,0%	95,5%	66,9%	67,3%	66,5%	94,6%	56
	640 x 640	96,7%	97,2%	96,3%	67,9%	68,2%	67,7%	95,0%	54
	896 x 896	96,7%	97,1%	96,3%	67,8%	67,7%	67,8%	95,3%	33

el tiempo de respuesta y los resultados obtenidos, se optó por usar una resolución de 640 x 640 para el entrenamiento de Scaled-YOLOv4-p5 sobre HHW.

En cuanto a las Figura 4 y Figura 5, se muestran la cantidad de instancias y las dimensiones de los *anchors* obtenidos mediante de k-means++ aplicado sobre un conjunto de puntos que representan el ancho y alto de cada una de las instancias del conjunto de entrenamiento para las bases de datos SHWD y HHW, respectivamente. Como se puede observar en Figura 4, gran parte de las instancias de SHWD son de muy baja resolución (< 50 x 50 píxeles), no obstante, también posee instancias de alta resolución que cubren la mayor parte de la imagen. En comparación con HHW (ver Figura 5), se observa que las instancias más grandes ocupan alrededor de la mitad de las imágenes y que se encuentran mejor distribuidas las cantidades de instancias entre los *anchors*. Otro aspecto importante por destacar es que en SHWD se observan 10 veces más instancias que en HHW, a pesar de que ambas bases de datos tienen aproximadamente la misma cantidad de imágenes. Por lo tanto, como se refleja en los resultados, la base de datos SHWD resulta más compleja para la tarea de detección producto a la mayor cantidad de instancias presentes en cada imagen y también a las dimensiones de las mismas.

Con respecto a los resultados cualitativos, se presentan algunas imágenes que ilustran el comportamiento de Scaled-YOLOv4-p5. En la Figura 6, se observa que Scaled-YOLOv4-p5 logra un correcto comportamiento sobre datos con múltiples instancias de los objetos de distintos tamaños y también con presencia de oclusiones. Por otro lado, cuando se analizan los resultados desfavorables obtenidos (Figura 7 y Figura 8) se puede observar

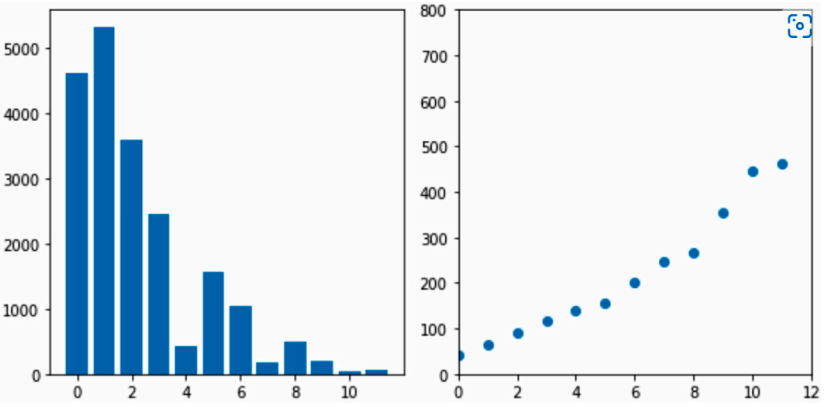

Figura 5. Gráficos que muestran la cantidad de instancias (izquierda) y la raíz cuadrada de las proporciones de los *anchors* (derecha) asociadas a cada *cluster* obtenido sobre el conjunto de entrenamiento de HHW.

Figura 6. Ejemplo de detecciones completamente correctas obtenidas sobre el conjunto de prueba de la base de datos SHWD.

Figura 7. Ejemplo de detección con baja precisión obtenida sobre el conjunto de prueba de la base de datos SHWD. La imagen de la izquierda contiene las anotaciones originales y la de la derecha a las predichas.

Figura 8. Ejemplo de detección con presencia de FPs y FNs sobre el conjunto de prueba de la base de datos HHW. La imagen de la izquierda contiene las anotaciones originales y la de la derecha a las predichas.

que las detecciones erróneas producidas por el método propuesto realmente no lo son, sino más bien corresponden a instancias de los objetos que no han sido correctamente anotadas. También se puede notar que los FNs producidos generalmente son sobre objetos pequeños en que la persona usa otro objeto en la cabeza (ej. un gorro) lo cual es confundido por el modelo como casco de seguridad.

CONCLUSIONES

En el presente trabajo de detección de cascos de seguridad, se evalúa un nuevo modelo de la familia YOLO, aplicando la detección en ámbitos de seguridad laboral de las industrias, y se comparan con otros modelos de detección propuestos. Para la evaluación y comparativa, se revisan varias bases de datos relacionadas con la detección de objetos en el marco de la seguridad laboral y se seleccionan dos bases de datos públicas apropiadas para el problema de detección binaria de cascos de seguridad. A partir de los resultados obtenidos, se demuestra que el método de detección de objetos Scaled-YOLOv4-p5 logra un rendimiento superior sobre estas dos bases de datos públicas en comparación con los otros métodos propuestos en estudios anteriores. Los resultados obtenidos son de 94,9% y 96,7% en términos de mAP@.50, y 93,8% y 95,0% en términos de F_1 para la base de datos SHWD y HHW respectivamente. Además, a partir del análisis cualitativo, se observa que muchos de los errores producidos por el modelo en realidad no son errores, sino que corresponden a anotaciones incompletas de los datos de las bases de datos originales. También, se muestra el comportamiento del modelo estudiado cuando se

entrena con imágenes de distintas resoluciones. Se observa que la decisión del tamaño de entrada depende de la base de datos objetivo, sin embargo, en todos los casos el modelo es posible ser usado en tiempo real con tarjetas gráficas de gama media o superior. Tenga en cuenta que uno de los factores atribuibles a la mejora en el rendimiento con respecto a estudios anteriores es la mejora en el diseño de la arquitectura Scaled-YOLOv4, que repercute en una reducción de los recursos computacionales y un escalado eficiente, lo que permite utilizar imágenes de mayor resolución que los estudios previos, facilitando así la detección precisa de pequeñas instancias de objetos que puedan estar presentes en las imágenes. Como trabajo futuro, se desarrollará un método basado en Scaled-YOLOv4 que logre diferenciar de mejor manera de cascos de seguridad con respecto a cualquier otro objeto ubicado en la cabeza, y que sea preciso y robusto en la detección binaria de cascos de seguridad sobre datos adquiridos en entornos más desafiantes que pueden presentar polución, oclusión, cambios bruscos de iluminación, entre otros factores.

AGRADECIMIENTOS

Los autores agradecen el apoyo de la Universidad Católica del Norte por el financiamiento parcial otorgado para este trabajo de investigación por medio del proyecto 202203010030-VRIDT-UCN.

REFERENCIAS

[1] Z. Wang, Y. Wu, L. Yang, A. Thirunavukarasu, C. Evison, and Y. Zhao, "Fast Personal Protective Equipment Detection for Real Construction Sites Using Deep Learning Approaches," *Sensors 2021,* vol. 21, no. 10, p. 3478, 2021, doi: 10.3390/S21103478.

[2] V.S.K. Delhi, R. Sankarlal, and A. Thomas, "Detection of Personal Protective Equipment (PPE) Compliance on Construction Site Using Computer Vision Based Deep Learning Techniques," *Frontiers in Built Environment*, vol. 6, p. 136, 2020, doi: 10.3389/FBUIL.2020.00136/BIBTEX.

[3] Aprueba guía para la selección y control de cascos de protección uso industrial, elaborada por el departamento salud ocupacional, Resolución-19 exenta, Ministerio de Salud, Subsecretaría de Salud Pública, Instituto de Salud Pública, Ene. 2013. [En línea]. Disponible: http://bcn.cl/2k229

[4] W. Fang, P.E.D. Love, H. Luo, and L. Ding, "Computer vision for behaviour-based safety in construction: A review and future directions," *Advanced Engineering Informatics*, vol. 43, p. 100980, 2020, doi: 10.1016/J.AEI.2019.100980.

[5] B. Zhao, H. Lan, Z. Niu, H. Zhu, T. Qian, and W. Tang, "Detection and location of safety protective wear in power substation operation using wear-enhanced YOLOv3 Algorithm," *IEEE Access*, vol. 9, pp. 125540-125549, 2021, doi: 10.1109/ACCESS.2021.3104731.

[6] H. Son, H. Choi, H. Seong, and C. Kim, "Detection of construction workers under varying poses and changing background in image sequences via very deep residual networks," *Automation in Construction*, vol. 99, pp. 27-38, 2019, doi: 10.1016/J.AUTCON.2018.11.033.

[7] B. Xiao, Q. Lin, and Y. Chen, "A vision-based method for automatic tracking of construction machines at nighttime based on deep learning illumination enhancement," *Automation in Construction*, vol. 127, pp. 103721, 2021, doi: 10.1016/J.AUTCON.2021.103721.

[8] 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, pp. 103085, 2020, doi: 10.1016/J.AUTCON.2020.103085.

[9] Q. Fang *et al.*, "Detecting non-hardhat-use by a deep learning method from far-field surveillance videos," *Automation in Construction*, vol. 85, pp. 1-9, 2018, doi: 10.1016/J.AUTCON.2017.09.018.

[10] 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*, 2020, doi: 10.1155/2020/9703560.

[11] H. Seong, H. Son, and C. Kim, "A Comparative Study of Machine Learning Classification for Color-based Safety Vest Detection on Construction-Site Images," *KSCE Journal of Civil Engineering,* vol. 22, Nº 11, pp. 4254-4262, 2018, doi: 10.1007/S12205-017-1730-3.

[12] L. Wang, L. Xie, P. Yang, Q. Deng, S. Du, and L. Xu, "Hardhat-Wearing Detection

Based on a Lightweight Convolutional Neural Network with Multi-Scale Features and a Top-Down Module," *Sensors 2020*, vol. 20, no. 7, p. 1868, 2020, doi: 10.3390/S20071868.

[13] G. Gallo, F. Di Rienzo, P. Ducange, V. Ferrari, A. Tognetti and C. Vallati, "A Smart System for Personal Protective Equipment Detection in Industrial Environments Based on Deep Learning," in *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*, Irvine, CA, USA, 2021, pp. 222-227, doi: 10.1109/SMARTCOMP52413.2021.00051.

[14] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[15] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 6517-6525, doi: 10.1109/CVPR.2017.690.

[16] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint, no*. 1804.02767, 2018, doi: 10.48550/arXiv.1804.02767.

[17] C.Y. Wang, A. Bochkovskiy, and H.Y.M. Liao, "Scaled-YOLOv4: Scaling Cross Stage Partial Network," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021, pp. 13024-13033, doi: 10.1109/CVPR46437.2021.01283.

[18] A. Bochkovskiy, C.Y. Wang, and H.Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection", *arXiv preprint,* no. 2004.10934, 2020, doi:10.48550/arXiv.2004.10934.

[19] G. Jocher *et al.* ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference. Zenodo. (2022). doi: 10.5281/ZENODO.6222936.

[20] D. Arthur and S. Vassilvitskii, "k-means++: The Advantages of Careful Seeding", in *SODA'07 Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007, pp. 1027-1035, doi: 10.5555/1283383.1283494.

[21] V. Isailovic *et al.*, "Compliance of head-mounted personal protective equipment by using YOLOv5 object detector," in *International Conference on Electrical, Computer, and Energy Technologies, ICECET 2021*, Cape Town, South Africa, 2021, pp. 1-5, doi: 10.1109/ICECET52533.2021.9698662.

[22] Njvisionpower, "Njvisionpower/Safety-Helmet-Wearing-Dataset: Safety helmet wearing detect dataset, with pretrained model," Accessed Jul. 07, 2022. [Online]. Available: https://github.com/njvisionpower/Safety-Helmet-Wearing-Dataset

[23] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, B. Leibe, J. Matas, N. Sere, M. Welling, Eds. vol. 9905, 2016, pp. 21-37, doi: 10.1007/978-3-319-46448-0_2.

[24] A.G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", *arXiv preprint,* no. 1704.04861, 2017, doi: 10.48550/arXiv.1704.04861.

[25] B. Kovács, A. D. Henriksen, J. D. Stets, and L. Nalpantidis, "Object Detection on TPU Accelerated Embedded Devices," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12899 , 2021, pp. 82–92, doi: 10.1007/978-3-030-87156-7_7/COVER/.

[26] Z. Li *et al.*, "Toward Efficient Safety Helmet Detection Based on YoloV5 with Hierarchical Positive Sample Selection and Box Density Filtering," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, 2022, doi: 10.1109/TIM.2022.3169564.

[27] B. Wang, H. Xiong, and L. Liu, "Safety helmet wearing recognition based on improved YOLOv4 algorithm," in *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Chongqing, China, 2022, pp. 1732-1736, doi: 10.1109/ITOEC53115.2022.9734707.

[28] H. Yu, Y. Sun, and C. Jia, "Safety Helmet Wearing Detection Based on Super-resolution Reconstruction," in *2021 IEEE 5th Information*

Technology,Networking, Electronic and Automation Control Conference (ITNEC), Xi'an, China, 2021, pp. 976-981, doi: 10.1109/ITNEC52019.2021.9587200.

[29] L. Zeng, X. Duan, Y. Pan, and M. Deng, "Research on the algorithm of helmet-wearing detection based on the optimized yolov4," *The Visual Computer*, pp. 1-11, 2022, doi: 10.1007/S00371-022-02471-9/TABLES/4.

[30] W. Jiang *et al.*, "Construction site safety detection based on object detection with channel-wise attention," in *ACM International Conference Proceeding Series*, 2021, pp. 85-91, doi: 10.1145/3511176.3511190.

[31] M. Sadiq, S. Masood, and O. Pal, "FD-YOLOv5: A Fuzzy Image Enhancement Based Robust Object Detection Model for Safety Helmet Detection," *International Journal of Fuzzy Systems*, pp. 1-17, 2022, doi: 10.1007/S40815-022-01267-2/FIGURES/17.

[32] H. Sun and P. Gong, "A Safety-Helmet Detection Algorithm Based on Attention Mechanism," *in 2021 7th IEEE International Conference on Network Intelligence and Digital Content, IC-NIDC 2021*, 2021, pp. 11–15, doi: 10.1109/IC-NIDC54101.2021.9660439.

[33] M. Zhou, J. Zhu and X. Li, "Safety helmet detection system of smart construction site based on YOLOv5S," in *2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP)*, Xi'an, China, 2022, pp. 1223-1228, doi: 10.1109/ICSP54964.2022.9778524.

[34] S. Farsiu, M.D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1327-1344, 2004, doi: 10.1109/TIP.2004.834669.

[35] S. Peled and Y. Yeshurun, "Superresolution in MRI: Application to Human White Matter Fiber Tract Visualization by Diffusion Tensor Imaging," *Magnetic Resonance in Medicine*, vol. 45, pp. 29-35, 2001, doi: 10.1002/1522-2594.

[36] A. Vaswani *et al.*, "Attention is all you need" in *31st International Conference on Neural Information Processing Systems*vol, 2017, 6000-6010, doi: 10.5555/3295222.3295349.

[37] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid and S. Savarese, "Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 658-666, doi: 10.1109/CVPR.2019.00075.

[38] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression," in *AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, 2020, pp. 12993-13000, doi: 10.1609/AAAI.V34I07.6999.

[39] H. Zhang, K. Zu, J. Lu, Y. Zou, and D. Meng, "EPSANet: An Efficient Pyramid Squeeze Attention Block on Convolutional Neural Network", *arXiv preprint,* no. 2105.14447, 2021, doi: 10.48550/arXiv.2105.14447.

[40] C.Y. Wang, H.Y.M. Liao, I.H. Yeh, Y.H. Wu, P.Y. Chen, and J.W. Hsieh, "CSPNET: a new backbone that can enhance learning capability of CNN," *in IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 390-391, doi:10.1109/CVPRW50498.2020.00203.

[41] S. Yun, D. Han, S. Chun, S.J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019, pp. 6022-6031, doi: 10.1109/ICCV.2019.00612.

[42] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011-2023, 2020. doi: 10.1109/TPAMI.2019.2913372.

[43] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137-1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition", *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904-1916, 2015, doi:10.1109/TPAMI.2015.2389824.

[45] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," *in IEEE conference on*

computer vision and pattern recognition, 2018, pp. 8759-8768, doi: 10.1109/CVPR.2018.00913.

[46] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," *in IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10778-10787, doi: 10.1109/CVPR42600.2020.01079.

[47] B. Xiao and S. C. Kang, "Development of an Image Data Set of Construction Machines for Deep Learning Object Detection," *Journal of Computing in Civil Engineering*, vol. 35, no. 2, p. 05020005, 2020, doi: 10.1061/(ASCE)CP.1943-5487.0000945.

[48] D. Peng, Z. Sun, Z. Chen, Z. Cai, L. Xie, and L. Jin, "Detecting Heads using Feature Refine Net and Cascaded Multi-scale Architecture", in *24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 2528-2533, doi: 10.1109/ICPR.2018.8545068.

[49] T.Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," in *European conference on computer vision. Lecture Notes in Computer Science*, vol. 8693, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. 2014, pp. 3686-3693, doi: 10.1007/978-3-319-10602-1_48.

[50] H. Zhang, M. Cisse, Y.N. Dauphin, and D. Lopez-Paz, "mixup: beyond empirical risk minimization", *arXiv preprint,* no. *1710.09412*, 2017, doi: 10.48550/arXiv.1506.02640.