Application of Deep-Learning Methods to Real Time Face Mask Detection

Diego González Dondo, Javier A. Redolfi, Daiana Garcia, and R. Gastón Araguás

***Abstract*—Due to the high rate of infection and the lack of a specific vaccine or medication for the new disease known as SARS-CoV2, the World Health Organization (WHO) has recommended the use of Personal Protective Equipment (PPE) as the main measure to avoid or reduce infections. One way to maximize compliance with this recommendation is through an automatic system that can recognize in real time whether a person is correctly using the corresponding PPE. This work presents the design, implementation and performance analysis of a system for recognizing the use of masks from image sequences, with the ability to operate in real time. Based on a generic object detection network, a training scheme is proposed for a detector of faces with masks and faces without masks, wherewith an average detection accuracy higher than 90% is obtained. This accuracy can be improved by using a network with a greater number of parameters, but with a longer computation time. The performance of the detector is validated with video sequences of people with and without facemasks, captured in different environments.**

***Index Terms*—Facemask detection, EPP detection, TinyYOLO, Neural Network, YOLOv3-tiny.**

I. Introducción

A finales de diciembre del año 2019 el Comité de Salud Municipal de Wuhan (China), reportó a la Organización Mundial de la Salud (OMS) que 27 personas habían sido diagnosticadas con neumonía, que hasta ese entonces el agente causal era desconocido. Hasta ese entonces, dentro de esas personas infectadas, se encontraban 7 individuos en estado crítico. La mayoría de estos casos eran trabajadores de un mercado de alimentos de la ciudad de Wuhan [1]. En poco más de un día, los agentes causales de neumonías como el SARS (Síndrome respiratorio agudo grave, por sus siglas en inglés), el MERS-CoV (Síndrome respiratorio por coronavirus de Oriente Medio, por sus siglas en inglés) y el virus de la gripe u otras enfermedades respiratorias causadas por virus, fueron descartados. Sin embargo, a partir del líquido de lavado bronco-alveolar de un paciente [2], una nueva cepa de coronavirus (2019-nCov) fue identificada y renombrada por el Comité Internacional de Taxonomía de Virus [3] como

D. González Dondo y G. Araguás pertenecen al Centro de Investigación en Informática para la Ingeniería (CIII) de la Universidad Tecnológica Nacional, Facultad Regional Córdoba, Argentina. e-mail: dgonzalezdondo@frc.utn.edu.ar, garaguas@frc.utn.edu.ar.

J. Redolfi es investigador del CIII de la UTNFRC y director del Grupo de Investigación sobre Aplicaciones Inteligentes (GISAI) de la UTN Facultad Regional San Francisco, Argentina. e-mail: jredolfi@sanfrancisco.utn.edu.ar

D. Garcia es investigadora del Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) de Argentina y trabaja en el Instituto de Investigaciones en Microbiología y Microtoxicología de la Universidad Nacional de Rio Cuarto, Argentina. e-mail: dgarcia@exa.unrc.edu.ar

Manuscript received April 19, 2005; revised August 26, 2015.

Fig. 1. Resultado de la aplicación de sistema propuesto sobre dos imágenes de prueba.

síndrome respiratorio agudo severo (SARS-CoV2). Basado en el rápido aumento en la tasa de infección humana, el día 11 de marzo la OMS clasificó al brote por SARS-CoV2 como una pandemia global. Esta enfermedad vírica respiratoria aguda provoca una mayor mortalidad en mayores de 60 años y en personas con afecciones previas como enfermedades cardiovasculares, enfermedades respiratorias crónicas, diabetes o cáncer [4]. Sobre la base de las pruebas actualmente disponibles, el virus se transmite de persona a persona mediante gotículas, fómites y el contacto directo con personas que tengan el virus (asintomáticas o no), y es posible que se transmita también a través de las heces. Hasta el momento, la OMS estima que la tasa de contagio (R0) del virus es de 1.4 a 2.5, aunque otras estimaciones hablan de un rango entre 2 y 3. Esto quiere decir que es altamente contagioso, donde cada persona infectada puede a su vez infectar entre 2 a 3 personas. Al tratarse de un virus nuevo (no más de 6 meses que se detectó), cuyo origen no está del todo claro todavía, como tampoco la evolución de la enfermedad que provoca, es recomendable adoptar la mayor cantidad de precauciones hasta que se disponga de más información [4]. Dada su alta tasa de contagio y a la falta de vacunas y medicación específica para esta patología, la OMS recomienda el uso de equipos de protección personal (EPP) para evitar el contagio. En especial para el personal sanitario que está en constante contacto con estos pacientes, entre otras enfermedades infectocontagiosas. Minimizar el riesgo de contagio entre los trabajadores de la salud con este tipo de virus es de especial importancia para evitar la reducción de este recurso humano tan necesario en presencia de una pandemia. El informe del Ministerio de Salud Argentina, bajo las recomendaciones dispuestas por la OMS, detalla que los camilleros, enfermeros, personal de limpieza, personal de RX, personal de laboratorio y médicos deben utilizar barbijo quirúrgico, camisolín, guantes y protección

ocular (además de la higiene de manos antes y después de estar con los pacientes). Los diferentes ministerios de salud y los cómites Ad-Hoc, para el tratamiento de la pandemia de distintas regiones y ciudades del mundo, en especial de América Latina donde se encuentra actualmente el foco de la pandemia, elaboran protocolos para la prevención de los contagios en la reapertura de los comercios e industrias. Estos protocolos de funcionamiento, siguiendo las recomendaciones de la OMS, exigen la utilización de protectores buconasales o tapabocas por parte del personal involucrado, distanciamiento social de al menos 2 metros entre las personas, entre otras medidas. Tal es el caso por ejemplo de las industrias automotrices de la República Argentina, particularmente las de la Provincia de Córdoba donde se les exige para proteger a su personal implementar en sus fábricas un paquete integral de medidas las cuales incluyen: toma de temperatura diaria al ingreso a las instalaciones, estrictas normas de higiene y limpieza, distanciamiento social y el uso en todo momento de protectores que cubren boca y nariz. Los conceptos de seguridad se aplican tanto en los puestos de trabajo como durante los descansos, en el almuerzo y en el viaje hacia las plantas industriales. Si bien las personas admiten fácilmente que el correcto uso de los diferentes equipos de protección personal (EPP) son de gran utilidad para disminuir los riesgos del contagio por SARS-CoV2, el uso continuo y prolongado de estos elementos suele generar incomodidad al trabajar. Una forma de garantizar su uso y prevenir el contagio por este patógeno en el sistema sanitario o en otros ambientes de trabajo es mediante un sistema de control inteligente que mediante un monitoreo permanente pueda generar una alarma o inhibir el acceso a una determinada área al detectar un uso indebido, o no uso, de estos EPPs. En este trabajo se describe el diseño, implementación y prueba de desempeño de un sistema de reconocimiento automático del uso de EPP a partir de secuencias de imágenes. En la Fig. 1 se puede ver un ejemplo de aplicación del sistema propuesto sobre dos imágenes de prueba. El artículo se organiza de la siguiente manera. En la sección II se presenta el problema a resolver, con sus posibles variantes e inconvenientes y se detallan las contribuciones del trabajo. En la sección III se describe el método propuesto, detallando los modos de entrenamiento utilizados considerando diferentes conjuntos de datos de acceso público y propios. La sección IV presenta los resultados obtenidos para cada modo de entrenamiento y en la sección V se consignan las conclusiones y los desafíos a futuro que se desprenden de este trabajo.

II. Planteamiento del Problema

La detección automática de objetos es una tarea fundamental en el área de visión por computadora y al mismo tiempo una de las más desafiantes. En la actualidad la gran mayoría de los detectores se basan en técnicas de aprendizaje profundo [5] y algunos de los métodos más utilizados para la detección de objetos son Faster R-CNN [6], SSD [7] y YOLOv3 [8]. La arquitectura Faster R-CNN [6] está planteada como dos redes que comparten los pesos de las capas iniciales, la primera de las redes es la encargada de detectar en que regiones de la imagen es problable que existan objetos y la otra es la encargada de calcular la probabilidad de que existe un objeto particular en cada región, combinando estos dos resultados se obtienen las regiones con objetos y las probabilidades de cada clase. Estas redes si bien comparten algunos pesos de las primeras capas, se conocen como redes de doble etapa y generalmente son más costosas. En cambio SSD [7] (del inglés detector de disparo simple) y YOLOv3 [8] (del inglés sólo miras una vez) se conocen como redes de etapa simple en donde con una red simple se obtienen tanto las regiones con objetos y la probabilidad de que tipo de objeto existe en cada región. Estas últimas redes al ser de simple etapa tienen ventaja con respecto al costo computacional. Si bien existen otras redes para detección de objetos que dan mejores resultados que las anteriores, como por ejemplo FCOS [5] o RetinaNet [9], para lograr funcionamiento en tiempo real se necesitan equipos de cómputo muy poderosos por lo que no se consideraron para este trabajo. A pesar de los remarcables progresos de los últimos años, la detección de caras en ambientes generales se mantiene como un problema abierto. Considerada como un caso especial de la detección de objetos genéricos, los detectores de caras que son estado del arte heredan muchas de las técnicas de los detectores de objetos genéricos. Especialmente técnicas de la familia de detectores de doble etapa basados en propuestas de regiones y luego en la clasificación de las mismas. Una desventaja de estos detectores de doble etapa es que son muy costosos computacionalmente, por esto para casos de funcionamiento en tiempo real los detectores de una sola etapa como SSD o Yolo ganan terreno [10]. Con respecto al estado del arte en la detección de caras, podemos nombrar a las redes RetinaFace [11] (basada en RetinaNet [9]), FDNet [12] o Face R-FCN[13] (estas últimas basadas en RCNN[14]). Si bien sus desempeños son muy buenos, una desventaja de estas redes es que para que funcione en tiempo real se necesitan GPUs (unidad de procesamiento gráfico, por sus siglas en inglés) muy potentes, las cuales tienen un costo muy elevado. En este trabajo se estudia la detección de caras con protectores buconasales o tapabocas, tarea que puede ser considerada como un caso particular de detección de caras con oclusiones. Este último problema, si bien está ampliamente investigado en la literatura, todavía está lejos de poder resolverse [15]. En particular, con respecto a la detección de protectores buconasales o tapabocas el único trabajo publicado que los autores han podido encontrar en la literatura presenta una comparación del estado del arte en detectores de caras en imágenes de salas de operaciones [16], y demuestra que se puede mejorar mucho la detección usando técnicas de auto-supervisión y datos no anotados. Sin embargo el objetivo que se plantean en [16] es detectar caras con protectores buconasales, pero sin distinguir entre caras con y sin protectores; tarea necesaria para realizar un control del correcto uso de dichos protectores. Una conclusión interesante a la que arriban los autores de ese trabajo es que los algoritmos actuales entrenados en imágenes naturales no generalizan bien para imágenes de salas de operaciones. Un trabajo similar es el de los autores del conjunto de datos MAFA (sección III-B1), en el cual el objetivo es encontrar caras con máscaras, pero nuevamente el objetivo no es distinguir entre caras con y sin

protector buconasal. El objetivo de este trabajo es diseñar un sistema para distinguir entre caras con y sin protector buconasal en videos en tiempo real. Hasta lo que conocemos no existe ningún trabajo en donde se proponga resolver el problema de distinguir entre caras con y sin protector buconasal. Un sistema de este tipo puede realizar un control en forma automática y de bajo costo del correcto uso de protectores buconasal en diferentes ambientes, permitiendo por ejemplo ayudar a los centros de salud, las diversas industrias y empresas a cumplir con los nuevos protocolos correspondientes a su actividad, y reducir así los riesgos de contagios de SARS-CoV2. Para resolver el problema presentado se propone reentrenar un detector de objetos genérico y planteamos como hipótesis fundamental que para obtener buenos resultados el detector debe ser reentrenado en cada ambiente de trabajo particular. El código utilizado, los modelos entrenados y los conjuntos de datos se encuentran publicados para su libre uso en un repositorio de GitLab del Centro de Investigación en Informática para la Ingeniería.

III. Solución Propuesta

A. Algoritmo de Detección

Si bien existen muchos detectores de caras en la literatura, los que mejores resultados dan necesitan de un gran poder de cómputo para lograr trabajar en tiempo real, lo cual es un problema si queremos implementar sistemas con muchas cámaras distribuidas en diferentes ambientes. En base al análisis realizado en el trabajo [16] en donde se concluye que detectores de objetos genéricos como SSD [7] o Faster-RCNN [6] dan muy buenos resultados en la detección de caras, y en base a un trabajo anterior [17] en donde comparamos diferentes redes neuronales convolucionales para la detección de EPPs en ambientes industriales, es que para este trabajo decidimos decidió utilizar la versión reducida de YOLOv3 [8], conocida como YOLOv3-tiny. El algoritmo YOLO (You Only Look Once, en inglés) es un sistema de detección de objetos en imágenes que, a diferencia de otros algoritmos que tienen dos etapas, hace uso de una única red neuronal convolucional para detectar objetos (de etapa simple). YOLO plantea la detección como un problema de regresión para predecir las regiones con objetos y las probabilidades para cada clase. Con una sola evaluación de la red se logra predecir las regiones y las probabilidades y además el entrenamiento se puede optimizar de extremo a extremo sobre la exactitud en la detección. A diferencia de los métodos antes mencionados que generan regiones con posibles objetos, luego clasifican las mismas y posteriormente refinan estos resultados, YOLO plantea la detección como un simple problema de regresión, directamente desde pixeles a regiones y probabilidades para las diferentes clases. El sistema divide la imagen de entrada en una grilla de $S \times S$, si el centro de un objeto cae dentro de la grilla de una celda, esa celda es la responsable de detectar a dicho objeto. En la Fig. 2 se muestra el tensor de salida generado por YOLOv3-tiny ante una imagen de entrada. Cada celda de la grilla genera una probabilidad de que haya un objeto $P(objeto)$, la posición (X, Y) del objeto y el ancho y alto del mismo; además se genera la probabilidad de que en caso de que haya un objeto este sea de una determinada clase, $P(CCB|objeto)$ para caras con protector buconasal y $P(CSB|objeto)$ para caras sin protector buconasal. Para detectar objetos con diferentes escalas y relaciones de aspecto, cada celda de la grilla luego genera B regiones de diferentes tamaños y relaciones de aspecto y para cada una de ellas predice la probabilidad de que haya un objeto, la posición del mismo y el ancho y alto de la región que lo contiene. En la Fig. 2 se muestra un vector de salida con dos regiones por celda indicadas en color azul y verde (Caja #1 y Caja #2). En la Fig. 3 se muestra un ejemplo con las regiones de detección que se aplican en cada celda. La principal ventaja de este método es que es extremadamente rápido. En este trabajo se utiliza la versión reducida de YOLOv3, conocida como YOLOv3-tiny, la cual tiene menos capas y por consiguiente menos parámetros, y una velocidad aproximadamente 5 veces mayor a la version completa, lo que le da la capacidad de correr en tiempo real incluso en máquinas sin GPU. En la Tabla I se observa la estructura de las capas que conforman la red YOLOv3-tiny.

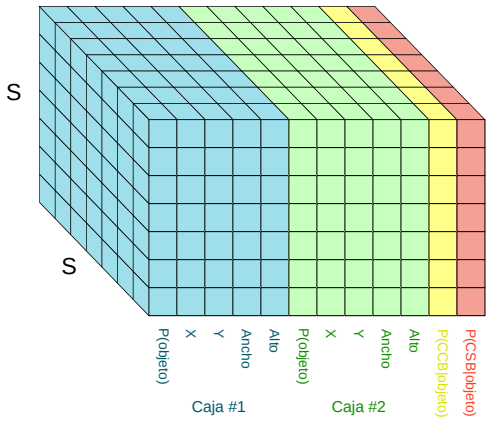


Fig. 2. Salida generada por la red YOLOv3-tiny para una imagen de entrada.

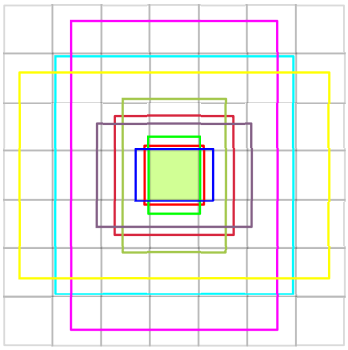


Fig. 3. Ejemplo con 9 regiones de detección de una celda para detectar objetos a diferentes escalas y relaciones de aspecto.

B. Conjuntos de Datos

Para el entrenamiento del detector es necesario contar con un conjunto de imágenes de entrenamiento, convenientemente etiquetadas con las clases que se quieren detectar. Idealmente, esta colección debe tener imágenes donde se observen rostros

TABLA I
ESTRUCTURA DE LA RED YOU ONLY LOOK ONE V3-TINY (YOLOV3-TINY).

Capa	Tipo	Filtros	Dim./paso	Entrada	Salida
0	Convolucional	16	3×3/1	416×416×3	416×416× 16
1	Maxpool		2×2/2	416×416×16	208×208× 16
2	Convolucional	32	3×3/1	208×208×16	208×208× 32
3	Maxpool		2×2/2	208×208×32	104×104× 32
4	Convolucional	64	3×3/1	104×104×32	104×104× 64
5	Maxpool		2×2/2	104×104×64	52×52× 64
6	Convolucional	128	3×3/1	52×52×64	52×52× 128
7	Maxpool		2×2/2	52×52×128	26×26× 128
8	Convolucional	256	3×3/1	26×26×128	26×26× 256
9	Maxpool		2×2/2	26×26×256	13×13× 256
10	Convolucional	512	3×3/1	13×13×256	13×13× 512
11	Maxpool		2×2/2	13×13×512	13×13× 512
12	Convolucional	1024	3×3/1	13×13×512	13×13× 1024
13	Convolucional	256	1×1/1	13×13×1024	13×13× 256
14	Convolucional	512	3×3/1	13×13×256	13×13× 512
15	Convolucional	255	1×1/1	13×13×512	13×13× 255
16	**YOLO**				
17	**Route 13**				
18	Convolucional	128	1×1/1	13×13 ×256	13×13 ×128
19	Sobre muestreo		2×2/1	13×13 ×128	26×26 ×128
20	**Route 19 8**				
21	Convolucional	256	3×3/1	13×13 ×384	13×13 ×256
22	Convolucional	255	1×1/1	13×13 ×256	13×13 ×256
23	**YOLO**				

o caras de personas utilizando y no utilizando protectores buconasal como mecanismo de protección. Luego de una revisión de los conjuntos de imágenes disponibles de acceso público no pudimos encontrar ningún con estas características, por lo tanto fue necesario construir uno empleando videos de internet y otros capturados en entornos industriales de nuestra región. Además de este conjunto de datos propio se utilizaron como complemento otros 2 importantes conjuntos de imágenes disponibles públicamente: uno con rostros humanos, WIDER FACE [18] y otro con rostros humanos ocluidos, MAFA [19]. Para el desarrollo de nuestro detector es necesario identificar solamente dos clases de objetos, caras con y sin protector buconasal, para lo cual se asignan las etiquetas "*CCB*" y"*CSB*" respectivamente. A continuación se describen los conjuntos de imágenes utilizados en este trabajo y su adaptación para el uso en los experimentos realizados.

1) Conjunto MAFA: El conjunto MAFA es una colección de 30 811 imágenes etiquetadas con 35 806 caras humanas ocluidas parcialmente con algún tipo de máscara. Cada etiqueta de la imagen cuenta con 6 atributos: a) ubicación de la cara en la imagen, b) ubicación de los ojos, c) ubicación de la máscara, d) orientación de la cara, e) grado de oclusión y f) tipo de máscara. En la Fig. 4 se pueden ver algunos ejemplos de las imágenes de este conjunto de datos. Para el presente trabajo, se adaptaron las anotaciones utilizando sólo la posición de la cara, con la consideración de tomar cada cara con oclusión como una cara con protector buconasal. De esta manera se construyó un nuevo conjunto de datos con las etiquetas "*CCB*" y "*CSB*", para las caras con y sin protectores buconasal respectivamente.

2) Conjunto WIDER FACE: Este conjunto de imágenes es una referencia para los algoritmos de detección de caras. Esta compuesto por 32 203 imágenes con 39 3703 caras de personas

Fig. 4. Ejemplos de imágenes del conjunto MAFA.

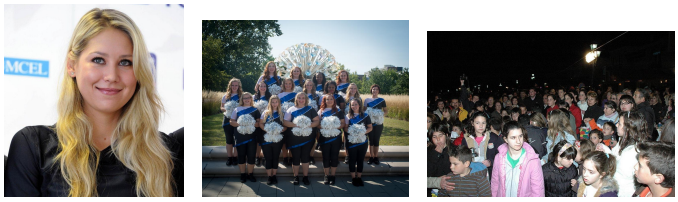


Fig. 5. Ejemplos de imágenes del conjunto WIDER FACE.

etiquetadas. La organización del conjunto de datos está basado en 61 tipos de eventos diferentes, tales como: partidos de fútbol, manifestaciones, conferencias de prensa, festivales, conciertos, entre otros. En este conjunto cada etiqueta contiene 7 atributos: a) ubicación de la cara en la imagen, b) nivel de distorsión, c) tipo de expresión, d) nivel de iluminación, e) nivel de oclusión, f) pose de la cara, y g) invalidez de la imagen. En la Fig. 5 se pueden ver algunos ejemplos de las imágenes de este conjunto de datos. La utilización de este conjunto requirió sólo la modificación del nombre de las etiquetas a "*CSB*", ya que las anotaciones referencian las posiciones de cada una de la caras en las imágenes.

3) Conjunto DEUBa: Con el objeto de verificar nuestra hipótesis principal, que el detector necesita ser entrenado en cada ambiente de trabajo en forma particular, se construyó un conjunto de datos con imágenes donde el entorno no presente grandes variaciones y que contenga caras de personas utilizando protectores buconasales. Este conjunto de datos fue elaborado utilizando 11 diferentes videos disponibles en la plataforma YouTube, en donde se muestran personas utilizando máscaras de protección en diversas situaciones y entornos: videos de trabajadores en las industrias, personas en la vía pública, entrevistas a personas y niños en las escuelas. De estos videos se formó un conjunto compuesto por 1 720 imágenes, con 1 261 caras con protectores buconasal y 993 caras sin protectores. Para su construcción se utilizó el software de etiquetado de imágenes Label Me, en donde cada etiqueta contiene solamente 2 atributos: a) utilización o no de protector buconasal y b) posición de la cara en la imagen. Se logró obtener de esta manera un conjunto de caras de tamaños heterogéneos y de diferentes resoluciones. En la Fig. 6 se pueden ver algunos ejemplos de las imágenes de este conjunto de datos. En la Tabla II se muestra una descripción detallada del conjunto de datos, donde para cada video utilizado se puede observar la cantidad de imágenes de entrenamiento y validación, la resolución de las mismas, la cantidad de objetos

Fig. 6. Ejemplos de imágenes del conjunto DEUBa.

para cada clase y el tamaño de los recuadros de los objetos.

C. Configuración Experimental

Para los experimentos se utilizó el framework Darknet el cual nos permite configurar los modelos, entrenarlos y evaluarlos. Los modelos de YOLOv3-tiny fueron configurados con un tamaño de entrada de 416×416, y con un máximo lote de entrenamiento de $50k$ (número de iteraciones). Se estableció un tamaño de lote de 32, con una subdivisión de 16, un momento de 0.9 y un decaimiento de 0.005. También se configuró la tasa de aprendizaje en 0.001 con una reducción de 0.1 veces del original a medida que se alcanzan los pasos de iteraciones $40k$ y $45k$ respectivamente. Para el entrenamiento de los distintos modelos se dividieron los diferentes conjuntos de datos en 2 partes en forma aleatoria: una parte formada por el $80\,\%$ de las imágenes del conjunto original para entrenamiento y otra por un $20\,\%$ para validación. En la Tabla III se muestra una comparación de los conjuntos empleados. Para la evaluación y comparación de los experimentos se utilizó como medida la precisión promedio media (mAP, por sus siglas en inglés). La mAP se mide como la media de la precisión promedio obtenida para cada una de las clases:

$$\mathrm{mAP} = \sum_{c=1}^{C} \mathrm{AP}_c \tag{1}$$

donde C es el número de clases y AP_c es la precisión promedio para la clase c. Esta precisión promedio o AP (por sus siglas en inglés) se calcula como el área bajo la curva de precisión y exhaustividad[1] en donde los positivos verdaderos se consideran cuando la intersección sobre la unión entre la caja limitante de la detección y la ubicación verdadera del objeto es mayor a 0.5, es decir:

$$\mathrm{IoU} = \frac{\text{Área de la Intersección}}{\text{Área de la Unión}} > 0.5 \tag{2}$$

Todos los entrenamientos fueron realizados en una PC de escritorio con un procesador AMD Ryzen 9 3900X 12-Core de $3.8\mathrm{GHz}$ con $16\mathrm{GB}$ de RAM y una GPU Nvidia Geforce RTX 2080Ti 11Gb.

D. Sistema de Detección Propuesto

Para la utilización del algoritmo en una aplicación real en una industria, se diseñó un sistema prototipo compuesto por una computadora industrial de placa simple[2] Intel NUC NUC8i3BEH con un procesador Intel Core i3-8109U de dos núcleos, con $8\mathrm{GB}$ de memoria RAM y una cámara Webcam Logitech C525 de $1\,280 \times 720$ pixeles de resolución. En esta PC se instaló un sistema operativo Debian versión 10.5 y el entorno Darknet. El sistema se completa con un monitor para visualizar las detecciones y una alarma sonora para indicar a las personas observadas cuando se comete una falta en el uso de los protectores. Este sistema prototipo se utilizó además para realizar las evaluaciones de tiempos de cómputo[3] y ver su factibilidad de utilización en tiempo real.

IV. Resultados

En esta sección se presentan los resultados de los diferentes experimentos propuestos para la evaluación del desempeño del sistema detector.

A. Configuración del Esquema de Entrenamiento

Para tener una aproximación inicial sobre la dificultad para detectar protectores buconasal planteamos una serie de experimentos que se describen a continuación.

1) Experimento I: Detector de Caras más Clasificador: El primer experimento consistió en entrenar un detector de caras y evaluar la capacidad del mismo para detectar caras con oclusiones. En el caso hipotético de que este resultado sea satisfactorio, como paso siguiente, se podría entrenar una pequeña red que clasifique las detecciones entre *CCB* y *CSB* para generar la salida final, de manera similar a la planteada en el trabajo [20] para clasificar señales de tránsito. Para esta evaluación inicial se entrenó YOLOv3-tiny sobre el conjunto de datos WIDER FACE (conjunto de imágenes que contienen sólo caras) y se evaluó su comportamiento sobre el primer video del conjunto de datos DEUBa descrito anteriormente, pero considerando las dos clases como una sola. Con esto se trata de evaluar la capacidad de un modelo entrenado para detectar caras descubiertas en la tarea de detectar caras con y sin protector buconasal. El resultado de este experimento fue una $\mathrm{mAP} = 0.61$, teniendo en cuenta que la precision promedio de detectores de caras es del orden del 0.9 [11], podemos concluir que el protector buconasal modifica la apariencia de una cara de tal manera que no es capaz de ser detectada por un algoritmo entrenado para detectar caras descubiertas. Cualquier intento de clasificación posterior entre *CCB* y *CSB* tiene como techo una mAP de 0.61; este valor no es suficiente para una aplicación real.

2) Experimento II: Detector de Caras y Protector Buconasal: Como segundo experimento se propone el entrenamiento de la red YOLOv3-tiny para detectar dos clases, *CSB* y *CCB*. Para la primera clase se utilizó el conjunto de datos WIDER FACE y para la segunda el conjunto MAFA. El resultado de este experimento tampoco fue satisfactorio, ya que se obtuvo una $\mathrm{mAP} = 0.60$, con una precisión media de un 0.69 para la clase *CCB* y de 0.50 para *CSB*.

[1]En lenguaje inglés esta curva es conocida como curva de precision/recall.
[2]Conocidas como Single Board Computers o SBC.
[3]Esta evaluación se realiza midiendo las FPS (frames por segundo) procesadas.

TABLA II
DESCRIPCIÓN DEL CONJUNTO DEUBA. TIPO DE AMBIENTE, CANTIDAD DE IMÁGENES DE ENTRENAMIENTO Y VALIDACIÓN, RESOLUCIÓN DE LAS MISMAS, CANTIDAD DE OBJETOS PARA CADA CLASE Y ESTADÍSTICAS SOBRE EL TAMAÑO DE LOS RECUADROS DE LOS OBJETOS.

Video	Descripción	N° de imágenes			Resolución		N° de objetos		Tamaño de los recuadros		
		Total	Entrenamiento	Validación	Ancho	Largo	CCB	CSB	Máximo	Mínimo	Promedio
Video1	Entorno Urbano 1	246	197	49	640.0	360.0	276	88	(117, 102)	(11, 17)	(51.0, 58.0)
Video2	Entorno Urbano 2	196	157	39	640.0	360.0	241	0	(339, 537)	(24, 29)	(119.0, 151.0)
Video3	Entorno Urbano 3	196	157	39	640.0	360.0	298	97	(326, 304)	(21, 28)	(69.0, 85.0)
Video4	Entorno Urbano 4	124	100	24	1920.0	1080.0	111	74	(165, 214)	(13, 12)	(69.0, 94.0)
Video5	Entorno Urbano 5	92	74	18	1080.0	1080.0	130	50	(200, 178)	(13, 18)	(58.0, 71.0)
Video6	Entrevista 1	27	22	5	1920.0	1080.0	23	6	(385, 469)	(29, 35)	(251.0, 342.0)
Video7	Entorno Fabril 1	114	92	22	480.0	360.0	86	40	(517, 769)	(26, 38)	(152.0, 196.0)
Video8	Entorno Fabril 2	321	257	64	400.0	224.0	246	128	(1296, 1070)	(43, 55)	(229.0, 297.0)
Video9	Escuela	91	73	18	1280.0	720.0	126	23	(158, 179)	(29, 37)	(64.0, 77.0)
Video10	Entorno Fabril 3	139	112	27	1920.0	1080.0	62	80	(169, 252)	(48, 59)	(106.0, 153.0)
Video11	Entrevista 2	174	140	34	1280.0	720.0	210	13	(333, 449)	(86, 121)	(152.0, 229.0)

TABLA III
CONJUNTOS DE DATOS UTILIZADOS. NÚMERO DE CLASES Y CANTIDAD DE MUESTRAS DE ENTRENAMIENTO Y EVALUACIÓN.

Conjunto de datos	**MAFA**	**WIDER**	**DEUBa**
Clases	1	1	2
Entrenamiento	2 4648	25 762	1 376
Evaluación	6 163	6 441	344

3) Experimento III: Detector de Caras y Protector Buconasal Entrenado en Ambiente de Uso: Este último experimento se plantea usando también dos clases pero con un único conjunto de datos de entrenamiento. El esquema consiste básicamente en entrenar el detector sobre el ambiente de trabajo en el que se va a utilizar. Se realizó un entrenamiento utilizando una parte del conjunto de datos DEUBa (Urbano 1, ver Tabla II) y evaluando sobre otra parte del mismo conjunto. Con este formato de entrenamiento se obtuvo una $\mathrm{mAP} = 0.89$, con una precisión promedio de 0.86 para la clase *CCB* y de 0.93 para *CSB*. Usando esta configuración se logró obtener una mejora de aproximadamente un $30\,\%$ con respecto a la propuestas anteriores, demostrando así la necesidad de entrenamiento en el ambiente de trabajo planteada inicialmente (sección II). Las precisiones obtenidas en estos tres experimentos se resumen en la Tabla IV, en donde también se detallan los conjuntos de imágenes utilizados para entrenamiento y validación.

B. Experimentos sobre DEUBa con Entrenamiento en el Entorno de Trabajo

Para validar la hipótesis planteada inicialmente que proponía que para obtener resultados satisfactorios era necesario entrenar en el entorno de trabajo final, también se realizaron experimentos sobre todos los videos del conjunto de datos DEUBa cuyos resultados se muestran en la Tabla V. En la Fig. 1 se muestran algunos resultados de la aplicación de YOLOv3-tiny sobre el conjunto de imágenes mencionado.

C. Experimento de Comparación con otras Redes

Se realizó una comparación de la versión reducida YOLOv3-tiny con la versión completa YOLOv3 y con la red Faster R-CNN. Estas redes se entrenaron y validaron utilizando el conjunto de imágenes *"Entorno Urbano 1"* de DEUBa, empleando un $80\,\%$ del conjunto para entrenamiento y un $20\,\%$ para validación. Para la red YOLOv3 se obtuvo una $\mathrm{mAP} = 0.93$ y para Faster R-CNN $\mathrm{mAP} = 0.953$. Para la evaluación del tiempo de ejecución se analizó la cantidad de FPS que se obtuvo al correr ambos modelos en la PC con y sin la utilización de la tarjeta gráfica (GPU). Con el empleo de YOLOv3-tiny se alcanzó una velocidad de procesamiento de 18 FPS para la detección de uso de protectores buconasal sin la necesidad de una tarjeta gráfica. En la Tabla VI se muestra la comparación de los comportamientos de los modelos analizados en CPU y en GPU: precisiones obtenidas para cada clase y los FPS logrados.

D. Evaluación Final sobre el Sistema Prototipo

El modelo YOLOv3-tiny se corrió sobre el sistema prototipo propuesto en III-D para la detección de caras con y sin protectores buconasal. La velocidad de procesamiento obtenida con este equipo estándar fue de 7 FPS, con lo cual se logra cumplir en forma satisfactoria la respuesta esperada para un sistema de alarma de uso de protectores buconasal. En la Fig. 7 se muestra una secuencia de imágenes del prototipo en funcionamiento. En la secuencia se observa una persona que ingresa al recinto sin protector buconasal, en el monitor del sistema se observa un sombreado en rojo sobre la imagen lo que indica que se detectó esta falta de uso. A partir de la tercera imagen la persona se coloca el protector buconasal, indicándose en el monitor del sistema con una leyenda en color verde.

V. CONCLUSIONES

En este trabajo se presentó el diseño, implementación y prueba de desempeño de un sistema de detección de personas usando protectores buconasal y personas que no lo están usando, para su utilización como sistema de control en ambientes donde el uso del protector buconasal se volvió obligatorio.

TABLA IV
PRECISIONES OBTENIDAS EN LOS EXPERIMENTOS SOBRE DIFERENTES CONJUNTOS DE DATOS.

Experimento	Conjunto de Entrenamiento	Conjunto de Evaluación	mAP	AP CCB	AP CSB
I	WIDER FACE	DEUBa Urbano 1 20 %	0.61	-	-
II	WIDER FACE & MAFA 80 %	DEUBa Urbano 1 20 %	0.60	0.69	0.50
II	DEUBa Urbano 1 80 %	DEUBa Urbano 1 20 %	0.89	0.86	0.93

Fig. 7. Imágenes del sistema prototipo en funcionamiento. En la secuencia se observa una persona que ingresa al recinto sin protector buconasal, en el monitor del sistema se observa un sombreado en rojo sobre la imagen lo que indica que se detectó esta falta de uso. A partir de la tercera imagen la persona se coloca el protector buconasal, indicándose en el monitor del sistema con una leyenda en color verde.

TABLA V
PRECISIONES OBTENIDAS DE LA APLICACIÓN DE YOLOV3-TINY SOBRE LOS CONJUNTOS DE DEUBA.

Video	Descripción	mAP	AP CCB	AP CSB
1	Entorno Urbano 1	0.89	0.86	0.93
2	Entorno Urbano 2	0.92	0.91	0.93
3	Entorno Urbano 3	0.90	0.92	0.88
4	Entorno Urbano 4	0.92	0.95	0.89
5	Entorno Urbano 5	0.91	0.94	0.88
6	Entrevista 1	0.93	0.90	0.96
7	Entorno Fabril 1	0.81	0.92	0.70
8	Entorno Fabril 2	0.98	0.96	0.99
9	Escuela	0.92	0.85	0.99
10	Entorno Fabril 3	0.95	0.92	0.98
11	Entrevista 2	0.94	0.88	1.00

TABLA VI
COMPARACIÓN DE LAS PRECISIONES Y FPS OBTENIDAS CON LAS REDES ANALIZADAS SOBRE *"Entorno Urbano 1"*.

		AP		FPS	
Red	mAP	CCB	CSB	GPU	CPU
Faster-RCNN	**0.95**	0.91	0.98	5	–
YOLOv3	0.93	0.92	0.94	105	5
YOLOv3-tiny	0.89	0.86	0.93	**240**	**18**

Hasta lo que sabemos, esta es la primera vez en la literatura que se investiga esta problemática. La hipótesis planteada inicialmente fue que para obtener una buena exactitud en la detección era necesario reentrenar la red en el lugar de trabajo final del detector; dicha hipótesis fue validada a través de los experimentos en donde demostramos un gran aumento en la precisión. Utilizando YOLOv3-tiny obtuvimos una mAP promedio en diferentes videos de aproximadamente 90 %. Además con el prototipo de hardware de evaluación planteado en la sección III-C se obtuvo una tasa de procesamiento de 7 FPS lo que para la aplicación pensada cumple con los requisitos de funcionamiento en tiempo real. Utilizando la versión completa de YOLOv3 obtuvimos una mejora aproximada de 7 puntos porcentuales sobre uno de los videos, por eso en caso de que se requiera una mayor exactitud, es necesario poner una red más compleja pero a costo de un mayor poder de cómputo para lograr funcionamiento en tiempo real. Por último, la red YOLOv3 fue diseñada para detectar objetos en un número grande de clases, pero creemos que para un problema más simple como el de detección o desambiguación entre dos clases no es necesaria una red tan grande y con tantos parámetros. Por esto planteamos como trabajo a futuro diseñar una nueva red en base a YOLOv3-tiny con un número menor de parámetros pero tratando de mantener la exactitud en la detección.

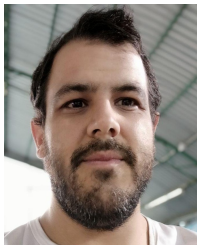
AGRADECIMIENTOS

Este trabajo fue desarrollado en el marco del proyecto homologo PID UTN - UTI3923TC "Detección de Objetos Usando Visión para Aplicaciones Industriales"de la Universidad Tecnológica Nacional, y el Convenio de Transferencia Tecnológica firmado con la empresa Caima Segall S.R.L. Los autores agradecen al Sr. Heraldo Romano, apoderado de la empresa mencionada por su contribución al proyecto.

REFERENCIAS

[1] ProMED-mail, "Coronavirus disease 2019 update (59): Global, cruise ship, who, promed-mail 2020; 28 marz:20200328.7153651; https://promedmail.org/promed-post/?id=20200328.7153651; consultado el 1 de abril de 2020." Technical documents, 2020.

[2] F. Wu, S. Zhao, B. Yu, Y.-M. Chen, W. Wang, Z.-G. Song, Y. Hu, Z.-W. Tao, J.-H. Tian, Y.-Y. Pei *et al.*, "A new coronavirus associated with human respiratory disease in china," *Nature*, vol. 579, no. 7798, pp. 265–269, 2020.
[3] C. S. G. of the International *et al.*, "The species severe acute respiratory syndrome-related coronavirus: classifying 2019-ncov and naming it sars-cov-2," *Nature Microbiology*, p. 1, 2020.
[4] O. M. de la Salud, "Prevención y control de infecciones para la gestión segura de cadáveres en el contexto de la covid-19: orientaciones provisionales, 24 de marzo de 2020," Technical documents, 2020.
[5] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9627–9636.
[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.
[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.
[8] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.
[10] J. Zhang, X. Wu, S. C. Hoi, and J. Zhu, "Feature agglomeration networks for single stage face detection," *Neurocomputing*, vol. 380, pp. 180–189, 2020.
[11] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-stage dense face localisation in the wild," *arXiv preprint arXiv:1905.00641*, 2019.
[12] C. Zhang, X. Xu, and D. Tu, "Face detection using improved faster rcnn," *arXiv preprint arXiv:1802.02142*, 2018.
[13] Y. Wang, X. Ji, Z. Zhou, H. Wang, and Z. Li, "Detecting faces using region-based fully convolutional networks," *arXiv preprint arXiv:1709.05256*, 2017.
[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.
[15] A. A. S. Alashbi and M. S. Sunar, "Occluded face detection, face in niqab dataset," in *International Conference of Reliable Information and Communication Technology*. Springer, 2019, pp. 209–215.
[16] T. Issenhuth, V. Srivastav, A. Gangi, and N. Padoy, "Face detection in the operating room: Comparison of state-of-the-art methods and a self-supervised approach," *International Journal of Computer Assisted Radiology and Surgery*, vol. 14, no. 6, pp. 1049–1058, 2019.
[17] M. Olmedo, J. A. Redolfi, D. González Dondo, and R. G. Araguás, "Evaluación Empírica De La Robustez De Diferentes Redes Neuronales Usadas Para La Detección De Objetos. XXIV Congreso sobre Métodos Numéricos y sus Aplicaciones-ENIEF 2019. Asociación Argentina. De Mecánica Computacional (AMCA). Santa Fe, Argentina, Noviembre 5-7," *Mecánica computacional*, 2019.
[18] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[19] S. Ge, J. Li, Q. Ye, and Z. Luo, "Detecting masked faces in the wild with lle-cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2682–2690.
[20] D. C. Santos, F. A. da Silva, D. R. Pereira, L. L. de Almeida, A. O. Artero, M. A. Piteri, and V. H. Albuquerque, "Real-time traffic sign detection and recognition using cnn," *IEEE Latin America Transactions*, vol. 18, no. 03, pp. 522–529, 2020.



Diego González Dondo Es Ing. en Electrónica y Dr. en Ingeniería por la Universidad Tecnológica Nacional. Es docente del Departamento de Ingeniería Electrónica de la UTN, Facultad Regional Córdoba e investigador en el área de fusión sensorial y visión por computadora en el Centro de Investigación en Informática para la Ingeniería en dicha Facultad.

Javier A. Redolfi Es Ing. en Electrónica y Dr. en Ciencias de la Ingeniería por la Universidad Nacional de Córdoba, Argentina. Actualmente es becario Post-Doctoral en la Universidad Tecnológica Nacional (UTN), docente e investigador en la misma Universidad. Además se desempaña como director del Grupo de Investigación sobre Aplicaciones Inteligentes (GISAI) de la UTN Facultad Regional San Francisco e investigador en el Centro de Investigación en Informática para la Ingeniería (CIII) para la Ingeniería de la UTN Facultad Regional Córdoba.

Daiana Garcia es Microbíologa, Magíster y Dra. en Ciencias y Tecnología Agraria y Alimentaria (doctorado Europeo). Actualmente es Investigadora Adjunta del CONICET la Universidad Nacional de Río Cuarto (URNC), docente e investigadora de las cátedras de Virología y Ecología Microbiana en la misma Universidad.

R. Gastón Araguás Es Ingeniero en Electrónica, y Doctor en Ingeniería por la Universidad Tecnológica Nacional de la República Argentina. Se desempeña como docente investigador en el área de la robótica y visión por computadoras. Desde el año 2018 es miembro de la Comisión de Posgrado de la UTN, y director del programa de Doctorado, mención Electrónica de la Facultad Regional Córdoba de la UTN. Es Profesor Asociado del departamento de Ingeniería en Electrónica de la Facultad Regional Córdoba de la UTN. Desde el año 2018 es director del Centro de Investigación en Informática para la Ingeniería (CIII) de la UTNFRC.