Journal
Boliviano de Ciencias

Proyecto de Ingeniería Aplicada

Identificación de personas sin barbijo utilizando Deep Learning

Identification of people without a mask using Deep Learning

Giovanny Germán Rocha Vallejo 1.

1. Licenciado en Informática. Universidad Privada del Valle. Cochabamba. Bolivia. giovo1@gmail.com.

RESUMEN

Este trabajo hace uso de las redes neuronales convolucionales para la detección de personas sin barbijo, ya que debido a la coyuntura actual del COVID-19 y de acuerdo con las medidas de bioseguridad instruidas por las instituciones gubernamentales y de salud, se ha visto de una manera comprobada que el uso de los barbijos o mascarillas quirúrgicas ayudan a reducir el riesgo de contagio de la enfermedad, por esta razón se hace evidente la necesidad de realizar la detección o identificación de personas que no estén utilizando un barbijo, incumpliendo con esta medida de bioseguridad y poniendo en riesgo a un grupo de la población.

Inicialmente se estableció un repositorio de entrenamiento compuesto por imágenes de personas con y sin barbijos, dichas imágenes habrían sido obtenidas de distintas fuentes.

Se han entrenado y comparado tres tipos de redes neuronales convolucionales, Faster R-CNN, SSD (Single Shot MultiBox Detector) y YOLO (You Only Look Once), cada una realiza la detección de personas con y sin barbijos, destacándose una de otra por su rapidez, precisión o rendimiento.

Para la obtención de los modelos de detección de objetos, se han utilizado los frameworks Darknet y TensorFlow Object Detection API, además de Google Colab que al ser un servicio de un proveedor gratuito, proveyó también potentes características computacionales.

Palabras clave: Visión por computador. Aprendizaje profundo. Red neuronal convolucional. Detección de objetos, YOLO.

Citar como: Rocha Vallejo, G. G. (2022). Identificación de personas sin barbijo utilizando Deep Learning. *Journal Boliviano de Ciencias*, 18(52), 34-44. https://doi.org/10.52428/20758944.v18i52.236

Editor invitado: Joaquin Humberto Aquino Rocha

Presidente comité científico COLEIC: Nahúm Gamalier Cayo Chileno

Revisado: 25/04/2022
Aceptado: 27/04/2022
Publicado: 29/06/2022

Declaración: Los autores declaran no tener ningún conflicto de intereses en la publicación de este documento.

ABSTRACT

This work makes use of convolutional neural networks to detect people with and without mask, due to the current situation of COVID-19 and in accordance with the biosecurity measures instructed by government and health institutions, it has been proven in a way that the use of surgical masks or chinstraps help to reduce the risk of contagion of the disease, the need to be able to detect or identify

people who are not wearing a mask becomes evident, not complying with this biosecurity measure and putting a group of the population at risk.

Initially, a training repository was established consisting of images of people with and without masks, these images were obtained from different sources.

Three types of convolutional neural networks have been trained and compared, Faster R-CNN, SSD (Single Shot MultiBox Detector) and YOLO (You Only Look Once), each one performs the detection of people with and without masks, standing out one from the other due to its speed, precision, or performance.

To obtain the object detection models, Darknet and TensorFlow Object Detection API frameworks have been used, Google Colab was used too, which, being a free provider, it also provided powerful computational features.

Keywords: Computer vision. Deep learning. Convolutional neural networks. Object detection. YOLO.

1. INTRODUCCIÓN

Una de las áreas de aplicación del Deep Learning es la detección de objetos, que es el proceso de identificación y localización de ciertas entidades que dependiendo del contexto interesa ubicar, ya sea que estén en una imagen o en un video. En la actualidad existen bastantes modelos de detección de objetos, que se diferencian de las novedosas técnicas que utilizan a la hora de realizar la detección, esto trae consigo que algunos modelos se destaquen por su precisión, rendimiento o velocidad a la hora de ser utilizados.

Las redes neuronales convolucionales son la base de los modelos de detección de objetos, si bien fueron creadas en los 90´s, hoy en día siguen apareciendo nuevas, cada una con mejores características que otra. YOLO (You Only Look Once) es una red neuronal convolucional que se destaca por su velocidad a la hora de realizar la detección de objetos, procesa toda una imagen una única vez, realizando la extracción, filtraje, localización y clasificación de los objetos con los que fue entrenada (Redmon et al., 2016).

Transfer learning es una técnica que permite utilizar el conocimiento obtenido de entrenar un modelo y utilizarlo en la obtención de otro. Con esto lo que se pretende lograr es una reducción del tiempo de entrenamiento así como obtener modelos más precisos y robustos (Pan y Yang, 2010). Fine tuning en deep learning implica el uso de los pesos de un algoritmo de aprendizaje profundo anterior para programar otro proceso de aprendizaje profundo similar (Joshi, 2020).

Los modelos de detección de objetos predicen un cuadro delimitador y una categoría de un objeto en una imagen. Intersection Over Union (IoU) se utiliza para determinar si el cuadro delimitador se predijo correctamente. IoU se define como el área de la intersección o superposición dividida por el área de la unión de un

recuadro delimitador predicho con un recuadro delimitador de verdad fundamental (Padilla et al., 2020)

Con la aparición del COVID-19 a finales del año 2019, el uso del barbijo se ha vuelto en algo básico para tratar de prevenir la enfermedad, es así que poder identificar a personas que no lo estén utilizando resulta algo útil y necesario, es por esta razón que se plantea en el presente trabajo utilizar una red neuronal convolucional entrenada para detectar personas sin barbijo.

2. DESARROLLO DE LA SOLUCIÓN

Para la obtención de los modelos de detección que permitan detectar personas con o sin barbijos se ejecutaron varias actividades, comenzando con la elaboración del repositorio de imágenes de entrenamiento, seguido de las herramientas y el entorno de trabajo utilizados, y finalmente el proceso de entrenamiento de las redes YOLO, Faster R-CNN y SSD.

1. Repositorio imágenes de entrenamiento

Para establecer el repositorio de imágenes de entrenamiento de la red neuronal, se hizo uso de cuatro Dataset públicos que contendrían imágenes de personas con y sin barbijos.

1.1. Unificación de los Dataset

Debido a que se utilizaron cuatro Dataset de distintas fuentes, para poder unificarlos en un solo repositorio de imágenes, se tuvo que realizar un preprocesamiento de las mismas junto con los archivos de texto que las acompañaban.

1.2. Generación de los archivos pivote

Una vez centralizadas y uniformizadas las imágenes y archivos de texto, se procedió a generar los archivos pivote del repositorio, que contendrían información de las clases u objetos a entrenar, así como también la ubicación y porcentaje de las imágenes que serían utilizadas para el entrenamiento y validación.

2. Herramientas y entorno de trabajo

Para la obtención de un modelo de detección utilizando YOLO se hizo uso de Darknet que provee muchas facilidades para el entrenamiento y validación de la red neuronal, así mismo se utilizó también Google Colab que al ser un entorno de trabajo en la nube, es gratuito y provee características computacionales de muy buen rendimiento. Para el entrenamiento de Faster R-CNN y SSD se utilizó el API de TensorFlow Object Detection.

3. YOLOv4

La arquitectura de la versión 4 de YOLO implementada en el framework de Darknet, está conformada por 162 capas, de convolución (conv), de ruta (route), de atajo (shortcut) y del tipo YOLO.

Para realizar el entrenamiento de la red neuronal, se tuvo que realizar algunas modificaciones a los archivos de configuración del framework, para que queden definidos valores como el número de clases a entrenar, número de filtros, tamaño del redimensionamiento de las imágenes, técnicas de data augmentation a utilizar.

3.1. Archivo con pesos pre-entrenados

Debido a que el uso de archivos con pesos pre-entrenados ayudan a que un detector de objetos personalizado sea mucho más preciso y no tenga que entrenar tanto tiempo, se procedió a descargar el archivo yolov4.conv.137[1] de pesos pre-entrenados para las capas convolucionales de la red neuronal YOLO v4.

3.2. Ejecución del entrenamiento

Para el proceso de entrenamiento, se habilitó el uso del GPU de Google Colab, lográndose completar todo el proceso en un tiempo total de 2 días.

4. Faster R-CNN y SSD

Para poder comparar los resultados obtenidos del entrenamiento realizado con YOLO v4, se realizó también el entrenamiento de las redes Faster R-CNN y SSD con el mismo repositorio de imágenes que fue utilizado por YOLO, pero se hizo uso del API de TensorFlow Object Detection para este propósito.

4.1. Conversión repositorio de entrenamiento

El API de TensorFlow Object Detección requiere que los archivos que serán utilizados para el entrenamiento y validación del modelo de detección estén en formato TFRecord, que es un formato serializado utilizado por TensorFlow para almacenar los datos. Es así que los archivos del repositorio tuvieron que ser convertidos del formato YOLO al formado TFRecord.

4.2. Modelos de detección pre-entrenados

Para el entrenamiento de las redes Faster R-CNN y SSD se utilizaron modelos de detección pre-entrenados, que permiten mejorar la precisión y tiempo de entrenamiento. Se hizo uso de los modelos SSD ResNet50 V1 FPN 640x640 (RetinaNet50) y Faster R-CNN ResNet50 V1 640x640 que están disponibles en (TensorFlow 2 Detection Model Zoo, 2021).

1 De https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.conv.137

4.3. Entrenamiento de Faster R-CNN y SSD

Debido a que Faster R-CNN y SSD requieren más ciclos de entrenamiento para obtener buenas detecciones, el tiempo total que fue requerido para completar este proceso fue de 5 días.

3. RESULTADOS

Después de haber entrenado las redes neuronales convolucionales YOLO v4, Faster R-CNN y SSD con el repositorio de imágenes de personas con y sin barbijos, a continuación, se presentan los resultados obtenidos seguidos del análisis respectivo.

1. Métricas obtenidas

En la Tabla 1 se muestran los resultados obtenidos de Verdaderos Positivos (VP), Falsos Positivos (FP) y Falsos Negativos (FN) por modelo:

Tabla 1. Valores de VP, FP y FN obtenidos

Modelo	VP	FP	FN
YOLO v4	1985	174	212
Faster R-CNN	1997	24	350
SSD	1809	25	538

Fuente: Elaboración propia, 2022.

Se puede observar que los tres modelos son capaces de detectar correctamente a la mayoría de los objetos. Es importante resaltar que el número de FP en YOLO es mayor con relación a Faster R-CNN y SSD, pero el número de FN es menor, por lo que se puede afirmar que YOLO logró detectar más objetos, aunque un grupo de estas detecciones no superaron el umbral IoU definido.

En cuanto a las principales métricas utilizadas para los modelos de detección se obtuvieron las que se muestran en la Tabla 2.

Tabla 2. Métricas obtenidas

Modelo	mAP@0.50	Precision	Recall	F1-Score
YOLO v4	90.99%	0.92	0.90	0.91
Faster R-CNN	88,9%	0.96	0.79	0.87
SSD	85.6%	0.95	0,74	0.83

Fuente: Elaboración propia, 2022.

Como se puede ver, los tres modelos tienen un mAP arriba de 85%, lo que significa que todos tienen un buen desempeño a la hora de realizar la detección de las clases entrenadas. En las Figuras 1 y 2 se muestran algunos ejemplos de las detecciones efectuadas.

Figura 1. Detecciones realizadas por Faster R-CNN

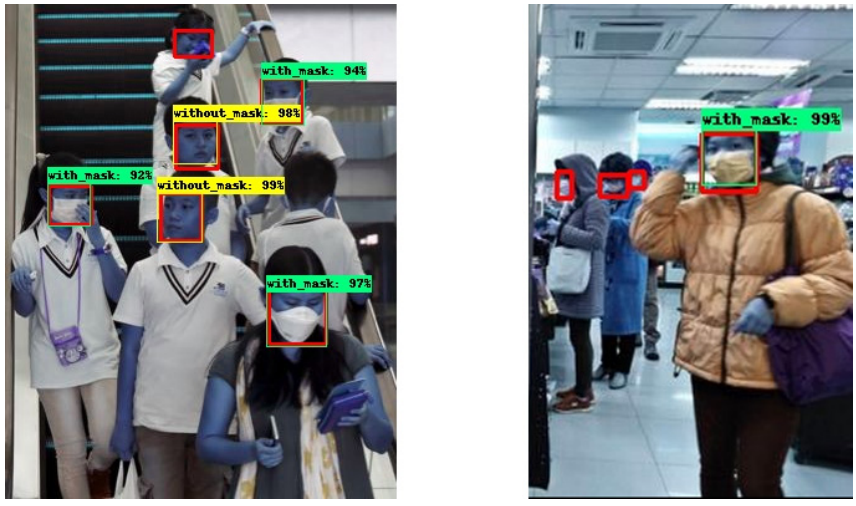
Fuente: Elaboración propia, 2022.

Figura 2. Detecciones realizadas por YOLO

Fuente: Elaboración propia, 2022.

Como se puede observar en la Tabla 2, Faster R-CNN y SSD obtienen mejor precision lo que significa que un buen porcentaje de las detecciones realizadas tienen realmente valor, pero su recall es más bajo, lo que indica que no lograron detectar a varios objetos relevantes como puede ver en la Figura 3, que es parte de algunas imágenes obtenidas del proceso de entrenamiento realizado.

Figura 3. Ejemplo objetos no detectados

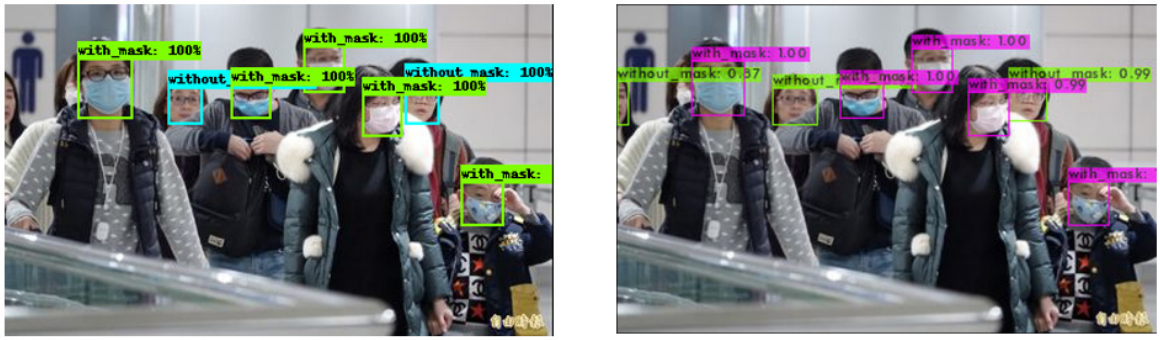


Fuente: Elaboración propia, 2022.

Los recuadros de color rojo son los recuadros delimitadores de verdad fundamental (ground truth bounding box), como se puede apreciar algunos no fueron detectados, pero los que sí lo fueron tienen un buen porcentaje de certeza.

En cuanto a *recall*, YOLO es el que tiene el valor más alto en esta métrica, lo que significa que logra detectar a un mayor número de todos los objetos existentes en una imagen, un ejemplo de aquello es mostrado en la Figura 4, donde se puede apreciar que YOLO detecta correctamente a una persona que está casi al límite de la imagen a diferencia de Faster R-CNN.

Figura 4. YOLO detecta una persona más que Faster R-CNN



Faster R-CNN **YOLO**

Fuente: Elaboración propia, 2022.

Así mismo, se puede ver también que, de los tres modelos, YOLO obtiene un F1-Score más alto, lo que significa que el modelo es más balanceado entre precision y recall, por lo que consigue obtener un buen número de detecciones efectivas y detectar a una mayoría de los objetos relevantes.

En cuanto a la velocidad a la hora de realizar las detecciones, para poder comparar los modelos, se utilizó FPS (Frame Per Second) que define que tan rápido un modelo de detección de objetos procesa un video y despliega los resultados obtenidos (Tabla 3).

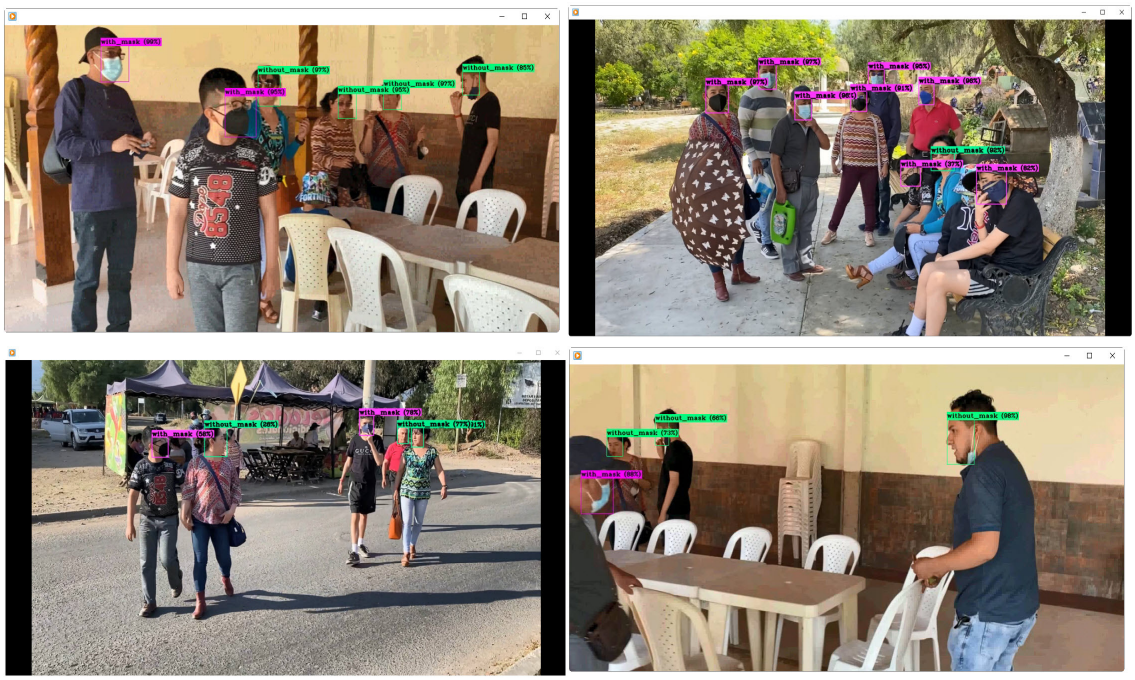
Tabla 3. Velocidad de los modelos

Modelo	FPS
YOLO v4	55.3
Faster R-CNN	10.5
SSD	16.9

Fuente: Elaboración propia, 2022.

Como se puede observar YOLO v4 obtuvo mejores resultados que Faster R-CNN y SSD, prácticamente realizando la detección en tiempo real, esta es una de las principales características de ese modelo, su velocidad. En la Figura 5 se muestran algunos ejemplos de las detecciones realizadas por YOLOv4.

Figura 5. Detecciones realizadas por YOLO en video



Fuente: Elaboración propia, 2022.

5. CONCLUSIONES

Se han utilizado tres de los principales modelos de detección de objetos que existen actualmente en el área del Deep Learning: Faster R-CNN, SSD y YOLO, para poder evaluarlos a la hora de realizar la detección de objetos, se procedió a entrenarlos, utilizando el mismo repositorio de imágenes de personas con y sin barbijos, se pudo evidenciar que algunos modelos requieren más tiempo y ciclos de entrenamiento que otros, como resultado final de todo ese proceso se pudo constatar que si bien Faster R-CNN y SSD tienen mejor precisión a la hora de realizar la detección, les falta mejorar la identificación de objetos relevantes en una imagen, así como también su velocidad a hora de realizar la detección. YOLO en cambio obtuvo muy buenos resultados en cuanto a la precisión e identificación de objetos relevantes en una imagen, logrando ser más equilibrado en ese aspecto, con relación a la velocidad a la hora de realizar la detección, fue sin lugar a duda muy superior a los otros modelos, realizando la detección prácticamente en tiempo real.

Como punto de partida para el proceso de entrenamiento de los tres modelos, se estableció un repositorio de imágenes de entrenamiento, el cual fue conformado por varios Datasets para tener mayor número de imágenes. Así mismo, debido a que se utilizaron distintas herramientas para el entrenamiento de los modelos, el repositorio de imágenes consolidado tuvo que ser readecuado al formato requerido por dichas herramientas.

Se ha podido observar que es posible la personalización de distintos tipos de redes neuronales convolucionales, en el caso del presente trabajo se habrían entrenado a las redes YOLO, Faster R-CNN y SSD a reconocer cuando una persona está utilizando un barbijo y cuando no, esto demuestra que es posible enseñar a una red neuronal convolucional a detectar cualquier tipo de objeto, por lo que la utilidad que se les podría dar es prácticamente en cualquier área.

Debido a que en la actualidad existen varios tipos de redes neuronales convolucionales, cada una con distintas características que las hacen adecuadas para distintos tipos de problemas, va a depender de los objetivos que se quieran lograr para seleccionar una u otra, ya que, si por ejemplo se requiere tener certeza al momento de detectar un objeto en vez de rapidez, utilizar YOLO no sería una buena opción.

REFERENCIAS

Bisong, E. (2019). Google Colaboratory. En Building Machine Learning and Deep Learning Models on Google Cloud Platform. Apress.

Bochkovskiy, A. (2021). GitHub - AlexeyAB/darknet: YOLOv4 / Scaled-YOLOv4 / YOLO - Neural Networks for Object Detection (Windows and Linux version of Darknet). Recuperado el 27 de Marzo de 2021, de GitHub: https://github.com/AlexeyAB/darknet

Bochkovskiy, A. (2021). Running a YOLOv4 Object Detector with Darknet in the Cloud! (GPU ENABLED). Recuperado el 27 de Marzo de 2021, de Google Colab: https://colab.research.google.com/drive/1_GdoqCJWXsChrOiY8sZMr_zbr_fH-0Fg

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv:2004.10934.

Girshick, R. (2015). Fast R-CNN. Proceedings of the IEEE International Conference on Computer Vision, (págs. 1440-1448).

Joshi, N. (2020). How to fine-tune your artificial intelligence algorithms. Recuperado el 20 de Abril de 2022, de Allerin: https://www.allerin.com/blog/how-to-fine-tune-your-artificial-intelligence-algorithms.

Khandelwal, R. (2019). COCO and Pascal VOC data format for Object detection. Recuperado el 28 de Marzo de 2021, de Towards Data Science: https://towardsdatascience.com/coco-data-format-for-object-detection-a4c5eaf518c5

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. (2016). SSD: Single Shot MultiBox Detector. European Conference on Computer Vision, (pp. 21-37).

Padilla, R., Netto, S. L., & da Silva, E. A. (2020). A Survey on Performance Metrics for Object-Detection Algorithms. International Conference on Systems Signals and Image Processing.

Pan, S. J., Yang, Q. (2010). A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, vol. 22, (pp. 1345-1359), doi: 10.1109/TKDE.2009.191.

Ponnusamy, A. (24 de Marzo de 2021). Preparing Custom Dataset for Training YOLO Object Detector. Obtenido de Vision Geek: https://www.visiongeek.io/2019/10/preparing-custom-dataset-for-training-yolo-object-detector.html

Redmon, J. (2013-2016). Darknet: Open Source Neural Networks in C. Obtenido de http://pjreddie.com/darknet/

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. IEEE.

TensorFlow 2 Detection Model Zoo. (1 de Mayo de 2021). Obtenido de GitHub: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

TensorFlow. (1 de Mayo de 2021). Obtenido de www.tensorflow.org

TensorFlow Object Detection API. (1 de Mayo de 2021). Obtenido de https://github.com/tensorflow/models/tree/master/research/object_detection

Fuentes de financiamiento: Esta investigación fue financiada con fondos del autor.

Declaración de conflicto de intereses: El autor declara que no tiene ningún conflicto de interés.