66

Fecha de presentación: marzo, 2023
Fecha de aceptación: mayo, 2023
Fecha de publicación: julio, 2023

COMPARATIVA

DE MODELOS DE DETECCIÓN DE OBJETOS Y PERSONAS EN ESPACIOS CERRADOS DE ACCESO PÚBLICO

COMPARISON OF DETECTION MODELS FOR OBJECTS AND PEOPLE IN CLOSED SPACES WITH PUBLIC ACCESS

Lester Yonabel Bográn Ortiz[1]
E-mail: lester.bogran@unah.edu.hn
ORCID: https://orcid.org/0000-0002-3178-8676
Jairo Jonathan Martínez Hernández[1]
E-mail: jairo.martinez@unah.edu.hn
ORCID: https://orcid.org/0000-0003-0004-6132
[1]Universidad Nacional Autónoma de Honduras. Honduras.

Cita sugerida (APA, séptima edición)

Bográn Ortiz, L. Y., Martínez Hernández, J. J. (2023). Comparativa de modelos de detección de objetos y personas en espacios cerrados de acceso público. *Universidad y Sociedad*, 15(4), 661-672

RESUMEN

Este trabajo tiene como objetivo presentar una comparativa de diferentes técnicas de detección de objetos y determinar cuál de ellos es el más adecuado para detectar personas en tiempo real y así controlar eficientemente el aforo de personas en espacios públicos cerrados para ayudar a prevenir la propagación del COVID-19. En este estudio se han puesto a prueba soluciones basadas en redes neuronales como R-CNN, YOLO y SSD, así como soluciones no neuronales como SVM y HOG. Se trabajo con modelos entrenados con del conjunto de datos MS COCO, y se utilizó el dataset Wisenet para evaluar los diferentes modelos. Todos los modelos fueron puestos a prueba en tres aspectos diferentes, siendo estos la precesión, velocidad de inferencia y el error. En las pruebas el modelo YOLO demostró un rendimiento mayor a los demás, mientras que la implementación de SSD demostró el rendimiento más bajo en mayoría de las pruebas.

Palabras clave: Detección de objetos, Visión por computador, Redes neuronales convolucionales, Aprendizaje Profundo, HOG.

ABSTRACT

This work aims to present a comparison of different object detection techniques and determine which of them is the most suitable to detect people in real time and thus efficiently control the capacity of people in closed public spaces to help prevent the spread of COVID-19. In this study, solutions based on neural networks such as R-CNN, YOLO and SSD, as well as non-neural solutions such as SVM and HOG, have been tested. We worked with models trained with the MS COCO dataset, and the Wisenet dataset was used to evaluate the different models. All models were tested in three different aspects, these being precession, speed of inference and error. In tests, the YOLO model demonstrated higher performance than the others, while the SSD implementation demonstrated the lowest performance in most tests.

Keywords: Object detection, Computer visión, Convolutional neural networks, HOG.

INTRODUCCIÓN

A casi dos años desde que el COVID-19 fue declarado pandemia (Ciotti, et al. 2020) y a pesar de los esfuerzos realizados, resulta muy difícil frenar su propagación. En muchos casos esto es debido a que las personas no suelen implementar correctamente las medidas de bioseguridad, no reconocen o tienen poco conocimiento sobre los riesgos de COVID-19 (Sarria et, al. (2021) haciendo que no utilicen apropiadamente las mascarillas y que no respeten el distanciamiento social.

Es este último él que gana el foco de interés de este trabajo. Lugares como supermercados, farmacias y bancos entre muchos otros, son establecimientos que deben atender una gran cantidad de personas diariamente con una concentración considerable. Esta aglomeración puede resultar en personas contagiando a otras de forma involuntaria al acercarse demasiado con otras personas. Para tratar de evitar estos contagios, algunos de estos establecimientos suelen restringir la cantidad de personas que pueden permanecer al mismo tiempo dentro de las instalaciones, llevando el control de las personas de forma manual, contando uno a uno las personas que ingresan y las personas que salen, técnicas como por ejemplo salen 5 entran 5 son utilizadas en algunos supermercados. En muchos casos esta técnica no es, necesariamente, precisa ni eficiente, ya que la persona encargada de llevar el control podría equivocarse en la cuenta o no estar atento en todo momento del ingreso de las personas.

Es en este punto en donde la tecnología, y en especial las técnicas de inteligencia artificial, pueden brindar una mano a la sociedad. Mediante el uso de herramientas de detección de objetos, por medio de cámaras de vigilancia, que por lo general suelen estar instaladas en este tipo de establecimientos, es posible identificar, entre muchas cosas, personas. Estas técnicas se pueden aprovechar para detectar y contar cuantas personas están dentro del establecimiento al mismo tiempo.

El problema que se está tratando en este trabajo es determinar cuál de las técnicas de detección de objetos tiene mejor rendimiento detectando personas en tiempo real para así ayudar a controlar el aforo de personas dentro de espacios cerrados de acceso público y así ayudar a reducir el contagio del COVID-19. Para cumplir con el objetivo planteado se realizó una búsqueda de los modelos de detección de objetos que trabajaran en tiempo real y los datasets más apropiados para trabajar con estos modelos y que fueran representativos del entorno en el cual se espera pueda ser aplicado un futuro proyecto de control de aforo.

Con el conjunto de datos seleccionado y los modelos entrenados se realizó una serie de pruebas para determinar cuál de los modelos es el que tenía el mejor comportamiento en tiempo real al momento de realizar la detección de personas.

En primer lugar, es importante definir que es la inteligencia artificial. Según McCarthy (2007) la inteligencia artificial se puede definir como la ciencia e ingeniería para fabricar maquinas que tengan inteligencia similar a la de los humanos, sin limitarse necesariamente a los comportamientos biológicos.

Por otro lado, Rich (1985) sostiene que la Inteligencia Artificial puede verse como el estudio de cómo lograr hacer que las computadoras hagan cosas en las que, de momento, las personas las realizamos mejor. En el caso particular de este trabajo, la detección de personas se puede de decir que, en efecto, los humanos son mejores que las computadoras identificado personas o cualquier tipo de objetos, por el momento. En la figura. 1 se puede apreciar la foto de un perrito que para una persona es fácilmente reconocible, sin embargo, una computadora solo verá una matriz de pixeles sin ningún contexto en particular.

Han transcurrido más 50 años desde que Larry Roberts, común aceptado como el padre de la visión por ordenador, discutiera, en la década de los 1960, la posibilidad de extraer información geométrica tridimensional partiendo de perspectivas bidimensionales. Mucho ha transcurrido desde ese momento y debido al amplio espectro de aplicaciones potenciales de la visión por ordenador, esta ha tenido que fusionarse con otros campos estrechamente relacionados como por ejemplo el procesamiento de imágenes. (Huang, 1996)

El campo de la visión por ordenador tiene diferentes áreas o subdominios, sin embargo, este trabajo se centra particularmente en la detección de objetos (object detection) (Zou et, al. 2019) ya que es de especial interés para llevar a cabo la comparativa que permitirá saber que modelo es más conveniente al momento de implementar un sistema de control de aforo.

Una de las tareas más importantes de la visión por computador es la de detectar objetos, la cual permite clasificar determinadas instancias que se encuentra en una imagen digital y asignarles una clase o categoría como personas, aminales o automóviles (Zou et, al. 2019). La detección de objetos es de gran relevancia ya que proporciona las bases para otras tareas de la visión por computador como, la segmentación de instancias, subtitulado de imágenes o seguimiento de objetos.

En la visión por computador, la detección de objetos se puede realizar por medio de diferentes técnicas como transformación de características invariantes de escala (del inglés

Scale Invariant Feature Transform SIFT), características robustas aceleradas (del inglés Speeded Up Robust Features SURF), o histograma de gradiente orientado (del inglés Histogram of Oriented Gradient HOG) entre otros (Li et, al. 2017).

Entre los modelos de detección de objetos es posible diferenciar entre los modelos con enfoque neuronal (es decir que implementan algún tipo de red neuronal para su funcionamiento) y los de enfoque no neuronal (estos son modelos que no implementan redes neuronales para su funcionamiento).

Dentro de estos modelos que no utilizan el enfoque neuronal se posible encontrar técnicas como SIFT, SURF, SVM (del inglés Support Vector Machines) o HOG (Pisner & Schnyer, 2019).

En cuanto a los modelos con enfoque basado en redes de neuronas existe una gran variedad de entre los cuales se pueden resaltar los modelos de redes neuronales convolucionales basadas en regiones (R-CNN) Fast R-CNN, Faster R-CNN (Ren et, al 2017). También existen otros modelos como Single Shot MultiBox Detector (SSD) (Liu et al., 2016), You Only Look Once (YOLO) (Redmon et, al. 2016), Single-Shot Refinement Neural Network for Object Detection (RefineDet), Retina-Net entre otros.

Existen muchas librerías o frameworks para trabajar con modelos de detección de objetos. Entre las más populares se encuentran OpenCV, TensorFlow, Torch/Pythorch, Scikit-learn, MXNet. En internet existen muchas más librerías a parte de las aquí expuestas.

Los modelos de detección de objetos que se listaron anteriormente debieron ser entrenados y puestos a prueba con un conjunto de datos o datasets. La creación de un dataset es una tarea que puede ser muy costosa, tanto en tiempo como en recursos. Internet se puede encontrar una gran variedad de datasets correctamente etiquetados para propósitos tales como la detección de objetos. Algunos ejemplos de estos datasets utilizados en el entrenamiento de modelos de detección de objetos son, Pascal VOC, ImageNet. Open Images Dataset. MS COCO (Lin et al., 2014; Marroquin et al., 2019) entre muchos otros. Estos últimos dos son los utilizados en este trabajo como conjunto de entrenamiento y validación respectivamente.

A.Datasets:

*Concebido como un conjunto de datos para apoyar la in*vestigación, MS COCO es un dataset que cuenta como más de 330 mil imágenes de las cuales más de 200 mil están etiquetadas. El objetivo de este dataset es avanzar en el estado del arte en el reconocimiento de objetos situando el reconocimiento de objetos en el contexto de la comprensión de la escena. Esto se logra mediante la recopilación de imágenes de escenas cotidianas que tienen objetos comunes en su contexto natural (Lin, et, al. 2014).

WiseNet es un conjunto de datos que proporciona una serie de videos de múltiples cámaras, así como de múltiples espacios con la información contextual completa del entorno. Este dataset, reagrupa 11 conjuntos de video (compuestos por 62 videos individuales) grabados usando 6 cámaras interiores colocadas en múltiples espacios. El dataset cuenta con pistas de 77 personas realizando diferentes acciones humanas como caminar, estar de pie o sentado, inmóvil, entrar o salir de un espacio y fusionar o dividir un grupo (Marroquin et al., 2019). Según sus autores, WiseNet es el primer dataset en proporcionar un conjunto de vídeos junto con la información completa del entorno, en este sentido, se muestra la ilustración en la figura 1.

Figura. No. 1. Ilustración de la posición de las cámaras para la generación del dataset WiseNet

Fuente: (Marroquin, Dubois, & Nicolle, 2019)

B. Metricas de desempeño

Existen diferentes métricas para evaluar el desempeño de los modelos de detección de objetos entre las más utilizadas están la precisión, la velocidad de inferencia, el consumo de memoria.

Entre los diferentes conjuntos de datos anotados utilizados por los desafíos de detección de objetos y la comunidad científica, existen diferentes métricas para medir la precisión de las detecciones realizadas, sin embargo, antes de revisar estas métricas, es importante resaltar algunos conceptos como verdaderos positivos (True Positive), falsos positivos (False Positive) y falsos negativos (False Negative).

Los verdaderos positivos se refiere a una detección correcta. Falsos positivos se refiere a una detección incorrecta de un objeto inexistente. Un falso negativo se refiere a un delimitador no detectado (Padilla et, al. 2020).

También es importante resaltar que en el contexto de la detección de objetos no se aplica los Verdaderos negativos (True Negative) ya que hay un infinito número de cuadros delimitadores que no deben detectarse dentro de una imagen determinada.

Es por esta razón que en detección de objetos se evita utilizar cualquier métrica que haga uso de los verdaderos negativos. Antes de continuar es importante definir que es una "detección correcta" para lo cual se analizará el concepto de intersección sobre la unión (Intersection Over Union IOU) el cual es una métrica de evaluación más popular utilizada en los puntos de referencia de detección de objetos (Rezatofighi, et al.,2019). En el ámbito de la detección de objetos, el IOU mide el área de superposición entre el cuadro delimitador predicho y el cuadro delimitador del real que debía ser detectado (ground truth) dividido por el área de unión entre ellos tal como se muestra en formula (1):

$$IOU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \qquad (1)$$

Donde:

B_p =es el cuadro predicho

B_{gt} =es el cuadro de ground truth

Al comparar el IOU con un umbral dado, se considera que una detección es correcta si IOU mayor o igual que dicho umbral e incorrecta en caso contrario.

Como se mencionó anteriormente el uso de verdaderos positivos no se utiliza en la detección de objetos y ninguna de las métricas que hagan uso de él. En lugar de esto se utilizan métricas como Precisión y Recall.

C. Precisión (Precision)

La precisión de un modelo se puede definir como la capacidad que tiene este para detectar solo aquellos objetos que son relevantes. La precisión de se puede definir en la formula (2):

$$P = \frac{TP}{TP+FP} = \frac{TP}{\text{todas las detecciones realizadas}} \qquad (2)$$

En otras palabras, la precisión es el porcentaje de detenciones positivas correctas.

D. Exahustividad(Recall)

La exhaustividad o recall se puede definir como la capacidad que tiene un modelo para encontrar todos los casos relevantes, en el caso de la detección de objetos sería identificar todos los elementos que se espera sean detectados para esto es necesario aplicar la formula (3)

$$R = \frac{TP}{TP+FN} = \frac{TP}{\text{ground truth}} \qquad (3)$$

En resumen, el recall es el porcentaje de predicciones positivas correctas entre todas las que debían ser detectadas.

En la detección de objetos, la velocidad de inferencia es una métrica que indica el tiempo que tarda un modelo en realizar las detecciones se mide en milisegundos. En esta métrica, métodos como SSD y YOLO suelen dar mejores resultados que métodos de dos etapas como los de la familia de R-CNN.

También existen las métricas de error. Para estimar los errores de los modelos de aprendizaje automático es común utilizar técnicas de medición de error como el error cuadrático medio (mean square error mse), raíz del error cuadrático medio (root square error rmse) o error absoluto medio (mean absolute error mae) (Chai & Draxler, 2014). Estas métricas se suelen utilizar en la clasificación y conteo de objetos.

E. Trabajos relacionados

Existen diversos trabajos relacionados con la detección de objetos en general y la detección de personas en particular. Como por ejemplo (2017) y (Gupta et al., 2021) en donde se aplica la detección de objetos para mejorar la seguridad o el trabajo de (Kumar et al., 2021) donde se realiza un sistema de conteo en centros comerciales utilizando SSD. También está el trabajo de Shulka et al. (2021) donde se implementa YOLOv5 para distinguir las personas presentes en un video. Todo lo anterior denota la importancia del tema de la detección de objetos y personas.

MATERIALES Y MÉTODOS

Para alcanzar los objetivos propuestos en este trabajo, se plantea la siguiente metodología dividida en las siguientes fases:

- Fase 1: Revisión exhaustiva de la bibliografía. En esta primera fase se realizará el estudio de la bibliografía enfocada en las diferentes técnicas de detección de objetos, las librerías que los implementan, así como los datasets que puedan brindar mejores resultados con los modelos que se seleccionen. De igual forma se revisará la bibliografía referente a trabajos relacionados con la detección de objetos y en especial la detección de personas en tiempo real. En esta fase se espera cumplir con el primer objetivo de esta comparativa.

- Fase 2: Análisis y selección de los dataset. En esta fase analizarán los datasets descritos en el estado del arte y se seleccionarán los más adecuados para este trabajo, en este caso es aquel que ayude más a la detección de personas. Al culminar esta etapa se estará cumpliendo con el segundo objetivo de este trabajo.
- Fase 3: Selección de los modelos que se pondrán a prueba. Una vez revisado el estado del arte se seleccionarán los modelos que prometan los mejores resultados en la detección de personas, para proceder a su implementación en el lenguaje de programación Python. Concluyendo esta fase se está cumpliendo con el tercer objetivo de este trabajo.
- Fase 4: Aplicar y comparar los modelos seleccionados. Con los modelos seleccionados se procederá a realizar las diferentes pruebas con los datasets seleccionados. Esta etapa ayuda a la consecución del cuarto y quinto objetivo.
- Fase 5: Análisis de los resultados. Para finalizar este estudio se realizará el análisis de los resultados obtenidos y se emitirán las conclusiones pertinentes. Es en esta fase en donde se verá plasmada la consecución del objetivo general de este trabajo.

Modelos de deteccion de objetos utilizados

Para la comparativa se los modelos puestos a prueba fueron Faster R-CNN, SSD, HOG, y YOLOv5. Estos modelos fueron seleccionados por tener según el estado del arte un buen rendimiento en la detección de objetos en tiempo real. de Faster R-CNN y SSD existen diferentes implementaciones disponibles en internet en sitios como TensorFlow Hub. De estas implementaciones se seleccionaron las implementaciones que fueran capaces de procesar más de 15 cuadros por segundo según su documentación oficial y con la mayor precisión posible. Para el caso de HOG se trabajó con la versión implementada en la librería Open CV. En cuanto a YOLO se trabajó con una versión YOLOv5s6 implementada en Pytorch Hub.

F. Conjunto de datos utilizados

En esta comparativa se utilizaron dos conjuntos de datos diferentes, uno para el entrenamiento de los diferentes modelos, el cual fue MS COCO por la variedad de imágenes de personas con las que cuenta. Para la validación de los modelos se utilizó el data set WiseNET el cual es un dataset conformado por varios conjuntos de videos generados por diferentes cámaras de video ubicadas en diferentes localidades de un edificio en donde las personas fueron grabadas realizando actividades cotidianas. Las cámaras utilizadas para la grabación de los videos que conforman el dataset Wisnet, cuentan con dos resoluciones diferentes. Un grupo de cuatro cámaras con una resolución de 1280X760 y otro grupo de siete cámaras con una resolución de 640X480.

G. Equipo utilizado

La realización de trabajos enfocados inteligencia artificial requiere de equipo de cómputo con cierte potencia que tiene costes monetarios bastante altos. Es por esta razón que para realizar esta comparativa el Equipo de cómputo utilizado fue una máquina virtual proporcionada por Google Coloboratory Pro, el cual proporciona una maquina con un procesador Intel Xeon a 2.30GHz 32GB de memoria RAM y una tarjeta gráfica Tesla P100 de 16GB de VRAM y un disco SSD de 200GB.

Para lograr los objetivos fue necesario realizar tres pruebas. La primera prueba fue la comparación de la velocidad de detección. Debido a que la intención de esta comparativa es ver que modelo de detección de objetos se desempeñaría mejor en un ambiente en tiempo real, una prueba en donde se mida el tiempo de respuesta de la detección resulta muy relevante.

La segunda prueba que se realizó fue la de comparación de la precisión. En esta prueba se comparó la precesión lograda por cada uno de los modelos de detección de objetos. El ganador de esta prueba fue el modelo que logró obtener la precesión más alta.

Finalmente, para la tercera y última prueba consistía en la comparación de errores. En este pequeño experimento se comparó el error en la detección de personas y aquel modelo con el error más bajo se considera que tiene un mejor desempeño.

Las tres pruebas que se plantean cuentan con elementos comunes que serán descritos en los siguientes apartados y posteriormente los elementos que son específicos de cada experimento. Entre los elementos comunes están:

- Factor de diseño: El factor de diseño para todos los experimentos es la detección de objetos. Para el factor de diseño se cuenta con cuatro niveles representado por cada uno de los modelos de detección de objetos que será evaluado en las diferentes pruebas. Estos se detallarán mejor en la siguiente sección. Los modelos que fueron evaluados son: HOG, SSD, Faster R-CNN y YOLO.
- Factores constantes: En esta comparativa existen dos factores que no varían en cada en cada nivel ni en las repeticiones. Estos factores son el conjunto de entrenamiento y el conjunto de evaluación. Estos se discutirán a continuación.

- Conjunto de datos de entrenamiento: Dada la disponibilidad de los modelos preentrenados con el conjunto de datos MS COCO, es este conjunto de datos el que se considera como el dataset de entrenamiento.
- Conjunto de datos de evaluación: Para el conjunto de evaluación se tomaron elementos de los datasets Wisenet para crear diferentes subconjuntos que serán utilizados en diferentes repeticiones.

Factores de molestia: Son elementos que pueden afectar el desarrollo del experimento pero que no pueden ser contralados. En este caso la limitada existencia de dataset etiquetados para el contexto en el que se desarrolla la comparativa. Otro factor de molestia son los posibles errores en el etiquetado de los datasets. Algunas imágenes podrían no tener todas las etiquetas que deberían tener o utilizar una región de etiquetado muy pequeña o grande, con relación a lo realmente necesitado, lo que podría repercutir de alguna forma en el rendimiento de los modelos al momento de ser evaluados ya que los elementos y regiones detectadas podrían diferir considerablemente. En la Tabla 1 se puede per la justificación del uso de estos datasets.

tabla.1. Descripción general de los conjuntos de datos entrenamiento y evaluación

Nombre	Descripción	Justificación
Conjunto de entrenamiento	Es el conjunto de datos que fue utilizado para entrenar los diferentes modelos que se están evaluando en esta comparativa	El conjunto de datos utilizado es MS COCO esto por la disponibilidad modelos preentrenados con este dataset
Conjunto de evaluación	Es el conjunto de datos que se utiliza para evaluar los diferentes modelos en esta comparativa	El conjunto de datos utilizado para la evaluación es Wisenet ya que proporciona un conjunto etiquetado en un ambiente similar al que se esperaría poder aplicar una implementación de estos modelos

Fuente: elaboración propia

Comparacion de velocidad

Como ya se mencionó antes, la primera prueba que se realizó es la comparación de la velocidad de los modelos. Para un sistema que funciona en tiempo real es deseable tiempos de respuesta cortos, es decir una alta velocidad de respuesta.

Comparacion de precisión

La segunda prueba que se realizó es la comparación de la precisión. Para conseguir esto, es necesario conocer los verdaderos positivos, los falsos positivos y los falso negativos obtenidos por cada modelo. Para obtener estos datos lo primero es calcular la intersección sobre la unión (IOU) que es la que permitirá saber si una detección es correcta o no.

Para considerar que una detección es correcta se debe establecer un umbral (threshold) que debe ser comparado con el IOU obtenido en una detección, sí el IOU es igual o mayor que dicho umbral entonces se considera que la detección es correcta (verdaderos positivos), en caso contrario, sí el IOU es menor que el umbral, entonces se considera que la detección es incorrecta (falso positivo). Los falsos negativos serán todos aquellos elementos que no se identificaron en la detección pero que si están presentes en la imagen.

Una vez encontrados lo, verdaderos positivos, falsos positivos y falsos negativos se puede calcular la precesión y el recall. En esta prueba cada uno de los modelos deberá detectar las personas que se encuentran una imagen. Para definir una detección como correcta se utilizó un umbral de 0.3 para comparar con el IOU.

Comparacion de error

En este último experimento se aplicaron métricas de error como el error cuadrático medio o el error absoluto medio. Estas métricas generalmente no se utilizan en la detención de objetos. Sin embargo, si se suelen emplear en el conteo de objetos y debido a que se espera detectar correctamente a las personas para así poder contarlas, es que se está tomando en cuenta esta métrica. Entre más bajo se el error obtenido, indicará que el desempeño del modelo es mejor.

RESULTADOS

Comparacion de velocidad

Una vez seleccionados los modelos entrenados se procedió a realizar las diferentes pruebas. La primera prueba fue una bastante sencilla en donde se midió el tiempo de inferencia de cada uno de los modelos. Como se describió anteriormente, se realizaron varias repeticiones para cada uno de los niveles o modelos. Estas repeticiones consistieron en realizar una inferencia sobre una imagen, es decir pedirle al modelo que indique cuantas personas hay en una imagen y donde concretamente se encuentran. Se realizaron 150 repeticiones, es decir que se seleccionaron 150 imágenes al azar del dataset de evaluación Wisenet. En esta prueba solamente se midió el tiempo que tarda un modelo en realizar la inferencia independientemente de si son correctas o no. Eso se revisará más adelante.

El primer modelo que se puso a prueba en esta comparativa fue la implementación de Faster R-CNN. Este modelo tuvo un tiempo mínimo de 1.4107 segundos y un tiempo máximo de 1.5472 segundos al realizar una inferencia. El tiempo total de ejecución fue de 224.0235 segundos para poder realizar la inferencia sobre 150 imágenes. El tiempo medio de inferencia fue de 1.4935 segundos. Algo importante a resaltar es que la primera inferencia que realiza el modelo tarda más que el resto de las inferencias por lo que, para los efectos de esta prueba, se realiza una primera inferencia que no se toma en cuenta, para permitir que los elementos del detector sean cargados correctamente. El resumen de estos datos se puede apreciar en la tabla 2.

Tabla. 2. Tiempo de inferencia del modelo Faster R-CNN

Medida	Tiempo en segundos
Menor tiempo de inferencia	1.4107
Mayor tiempo de inferencia	1.5472
Tiempo total de inferencia	224.0235
Tiempo medio de inferencia	1.4935

Fuente: Elaboración propia

La implementación de SSD seleccionada para esta prueba obtuvo como menor tiempo de inferencia 1.4691 segundos mientras que el mayor tiempo de inferencia fue de 1.5667 segundos. Para completar la inferencia sobre las 150 imágenes, este modelo se tomó 228.1001 segundos teniendo un tiempo medio de inferencia de 1.5207 segundos. En la tabla 3 se puede apreciar este resumen. Al igual que ocurría con el modelo anterior es necesario realizar una inferencia de prueba para realizar la carga de los elementos del modelo.

Tabla. 3. Tiempo de inferencia del modelo SSD

Medida	Tiempo en segundos
Menor tiempo de inferencia	1.4691
Mayor tiempo de inferencia	1.5667
Tiempo total de inferencia	228.1001
Tiempo medio de inferencia	1.5207

Fuente: Elaboración propia

El tiempo mínimo que le tomó al modelo HOG, realizar una inferencia fue de 0.6957 mientras que el tiempo máximo fue de 0.7887. En total le tomó 109.6666 realizar la inferencia a todas las imágenes. El tiempo promedio de inferencia es de 0.7311. A diferencia de los otros modelos esta implementación no requiere de una inferencia de prueba para la carga de los elementos. Los resultados se pueden apreciar en la tabla 4.

Para la implementación de YOLO el tiempo total de inferencia fue de 4.2081. Este modelo obtuvo un tiempo mínimo de inferencia de 0.0246 y el tiempo máximo de inferencia es de 0.0335. Yolo obtuvo un tiempo de inferencia medio de 0.0280. Estos valores se resumen en la Tabla 5.

Tabla. 4. Tiempo de inferencia del modelo HOG

Medida	Tiempo en segundos
Menor tiempo de inferencia	0.6957
Mayor tiempo de inferencia	0.7887
Tiempo total de inferencia	109.6666
Tiempo medio de inferencia	0.7311

Tiempo de inferencia del modelo YOLO

Medida	Tiempo en segundos
Menor tiempo de inferencia	0.0246
Mayor tiempo de inferencia	0.0335
Tiempo total de inferencia	4.2081
Tiempo medio de inferencia	0.0280

Fuente: Elaboración propia

Comparacion de precisión

Esta es la medida más compleja de esta comparativa ya que, como se mencionó en apartados anteriores, se debe realizar una serie de pasos. El primer paso es estimar el IOU de cada uno de los elementos detectados en una

inferencia para posteriormente determinar los verdaderos positivos, falsos positivos y falsos negativos.

Uno de los principales problemas que se encontró en esta prueba fue la doble detección en donde una misma caja en el ground truth era detectada correctamente más de una vez. Para solucionar este problema se tomó como la única detección correcta para esa caja en particular a aquella que tuviera el IOU más alto. Sin embargo, antes de descartar el resto como falsos positivos se verificó si la misma caja no coincidía con otra detección en el ground truth así se aseguró que las detecciones tomadas como correctas fueran las más apropiadas para cada caja del ground truth

Con estos valores obtenidos se pudo hacer el cálculo de la precisión y la exhaustividad (recall). Estos pasos se realizaron para cada uno de los modelos. Este procedimiento es general para cada uno de ellos por lo que en los siguientes apartados únicamente se mostraran los resultados obtenidos de cada uno de ellos.

La implementación de Faster R-CNN obtuvo un total de 257 detecciones de las cuales 176 fueron catalogadas correctamente o verdaderos positivos y 83 fueron catalogadas de forma incorrecta o falsos positivos. Del total de 210 detecciones que debía realizarse, hicieron falta 36 las cuales son consideradas como falsos negativos. Aplicando la fórmula de la precisión, sobre los resultados anteriores, Faster R-CNN obtuvo un valor de 0.6770, mientras que con la fórmula de la exhaustividad obtuvo un valor de 0.8286. El modelo de SSD obtuvo un total de 512 detecciones de las cuales 201 eran verdaderos positivos y 311 resultaron ser falsos positivos. Este modelo obtuvo 9 falsos negativos. Con estos datos, SSD obtuvo una precesión de 0.3926 y una exhaustividad de 0.9571. El modelo de HOG contó con un total de 312 detecciones de las cuales solo 63 resultaron ser verdaderos positivos y 249 fueron catalogadas como falsos positivos. 147 detecciones no fueron realizadas correctamente por este modelo. HOG obtuvo una precisión de 0.2019 y una exhaustividad de 0.3. Yolo realizó un total de 186 detecciones, de las cuales, 169 fueron realmente verdaderos positivos y 17 como falsos positivos haciendo que hicieran falta 41 detecciones. La precisión obtenida por este modelo es de 0.9086 y la exhaustividad es de 0.8048.

Comparacion de error

Esta es una métrica que no suele utilizarse en el contexto de la detección de objetos y es más utiliza en el contexto de la clasificación o el conteo. Es justamente por esto último que en este trabajo se está tomando en cuenta. Para el cálculo de errores existen diferentes métricas como el error cuadrático medio o el error absoluto medio. Para realizar esta comparativa únicamente se tendrá en cuenta los valores predichos por cada modelo en cada imagen y los valores esperados en cada imagen. En los siguientes apartados se detallan los valores obtenidos por estas métricas aplicados sobre cada uno de los modelos.

El resultado de aplicar las métricas de erro sobre las inferencias realizadas por Faster R-CNN indican un error cuadrático medio de 1.0867. Una raíz de error cuadrático medio de 1.0424 y por último un error absoluto medio de 0.6333. Para las inferencias de SSD los resultados sobre los diferentes errores son los siguientes. El error cuadrático medio es de 7.6533 mientras que la raíz de este da como resultado un valor de 2.7665. Por último, el resultado del error absoluto medio es de 2.04. HOG obtuvo un error cuadrático medio de 2.32 mientras que la raíz de este da como resultado un valor de 1.5232. El valor absoluto del error medio es de 1.1867. YOLO consiguió un valor de 0.2267 lo que al error cuadrático medio se refiere, mientras que la raíz de este da como resultado un valor de 0.4761. Para finalizar, la implementación de YOLO utilizada en este trabajo obtuvo un error absoluto medio de 0.2133.

DISCUSIÓN

En esta sección se discuten los resultados obtenidos en las diferentes pruebas realizadas a cada uno de los modelos comparados en este trabajo. En primer lugar, se discutirán los resultados de la prueba de velocidad de inferencia, seguido del de los resultados de la prueba de precisión y finalizando con los resultados de la prueba de error.

Resultados de velocidad

En la prueba de velocidad se evaluó el tiempo que tardó cada uno de los modelos. En lo referente al tiempo total que le tomó a cada modelo realizar la inferencia sobré las 150 imágenes seleccionado los resultados indican que el modelo SSD tardó más de 224 segundos siendo este el modelo que más tiempo tomó para realizar la inferencia. En contraposición a lo anterior YOLO es el modelo que menos tiempo tardó en realizar la inferencia sobre todas las imágenes, con un tiempo total de 4.2081 segundos. En la tabla 5 se puede apreciar los tiempos totales obtenidos por cada uno de los modelos.

En cuanto al menor tiempo de inferencia, en la tabla 6 se puede apreciar que estos tiempos son consistentes con los resultados anteriores, siendo YOLO el modelo con menor tiempo de inferencia, mientras que el modelo SSD es el modelo más lento.

Tabla.5.Tiempo total de inferencia por modelo

Modelo	Tiempo en segundos
Faster R-RCNN	224.0235
SSD	228.1001
HOG	109.6666
YOLO	4.2081

Fuente: Elaboración propia

Tabla. 6. Tiempo menor de inferencia por modelo

Modelo	Tiempo en segundos
Faster R-RCNN	1.4107
SSD	1.4691
HOG	0.6957
YOLO	0.0246

Fuente: Elaboración propia

Con el tiempo mayor de inferencia, nuevamente se puede apreciar que los resultados son consistentes a los presentados anteriormente. En este caso el mayor tiempo de inferencia de YOLO es de 0.0335 y mientras que el tiempo de SSD es de 1.5667. Estos resultados se pueden apreciar en la tabla 7.

tabla.7. Tiempo mayor de inferencia por modelo

Modelo	Tiempo en segundos
Faster R-RCNN	1.5472
SSD	1.5667
HOG	0.7887
YOLO	0.0335

Fuente: Elaboración propia

Finalmente, los tiempos medios de inferencia por cada modelo tabla 8 nuevamente mantienen las relaciones de vistas en las comparaciones anteriores en donde YOLO tiene el menor tiempo de inferencia media de todos los modelos, mientras que SSD es el modelo con mayor tiempo de inferencia.

Tabla.8. Tiempo Medio de inferencia por modelo

Modelo	Tiempo en segundos
Faster R-RCNN	1.4935
SSD	1.5207
HOG	0.7311
YOLO	0.0280

Fuente: Elaboración propia

Los resultados de esta prueba indican que YOLO es el modelo con los mejores tiempos de inferencia, es decir que es que es el que menos tiempo tarda en realizar una inferencia sobre una imagen. Mientras que SSD es el modelo con los peores tiempos de inferencia, seguido muy de cerca del modelo Faster R-CNN. HOG por su parte obtuvo un resultado considerablemente mejor, comparado con los dos anteriores, sin embargo, sus resultados son aún distantes de los de YOLO. En la siguiente grafica se puede ver la comparativa. En la Figura 2 se puede muestra una comparativa de los tiempos de inferencia de cada modelo.

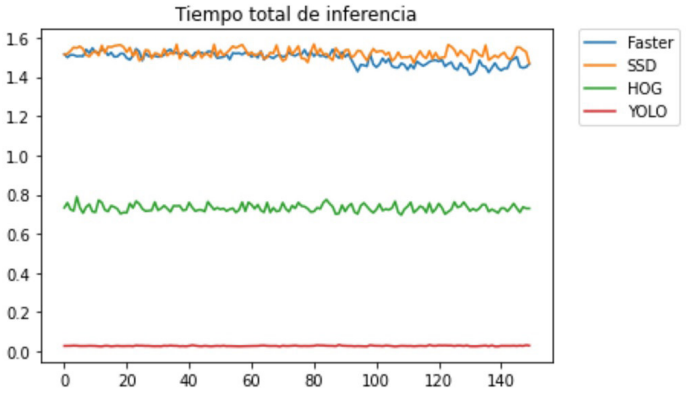

Figura. 2. Grafica de tiempos de inferencia de cada modelo por todas las imágenes. (fuente propia)

Resultados de precisión

Para calcular los valores de precisión es importante obtener los verdaderos positivos, falsos positivos, falsos negativos. Para todos los casos el total de detecciones que se esperaban (ground truth) eran 210. Para darles la mejor oportunidad a todos los modelos, en la estimación de una detección correcta, se utilizó un valor de IOU de 0.3 En cuanto a los verdaderos positivos SSD fue el modelo que más elementos identificó correctamente mientras y HOG fue el modelo que menos aciertos tuvo con una identificación apropiada solo 63 de las 312 detecciones realizadas.

En cuanto a la detección de los falsos positivos, también fue SSD el que tuvo el mayor número de detecciones incorrectas con un valor de 311 superando el número de detecciones correctas que este modelo realizó. Por su parte YOLO fue el modelo que menos detecciones incorrectas realizó, con un total de 17 falsos positivos.

Por último, el que menos falsos negativos obtuvo también fue el modelo SSD, con una total de 9 falsos positivos, seguido por Faster R-CNN con 36 y YOLO con 41. Por su parte HOG obtuvo un total de 149 falsos positivos, lo que

quiere decir que no fue capaz de detectar correctamente ni la mitad de las personas.

Estos resultados se resumen en la tabla 9, donde es interesante resaltar que todos los modelos detectaron más elementos de los presentes, excepto YOLO que detectó menos de los esperados.

Tabla.9. Resultados de verdaderos positivos(VP), falsos positivos(FP) y falsos negativos(FN)

Modelo	VP	FP	FN	Total Detectadas
Faster R-RCNN	174	83	36	257
SSD	201	311	9	512
HOG	63	249	147	312
YOLO	196	17	41	186

Fuente: Elaboración propia

Con los valores de la Tabla X es posible realizar el cálculo de la precisión. En donde se puede apreciar que YOLO es el que mayor precesión muestra, con un valor de más del 90% mientras que SSD a pesar de haber realizado más detecciones correctas que YOLO y haber omitido menos detecciones esperadas, solo obtiene una precisión de apenas el 39.26%. Sin embargo, en cuanto a precisión el modelo con los peores resultados es HOG con una precisión del 20.19%. En la Figura 4. se pueden apreciar la comparativa de estos resultados.

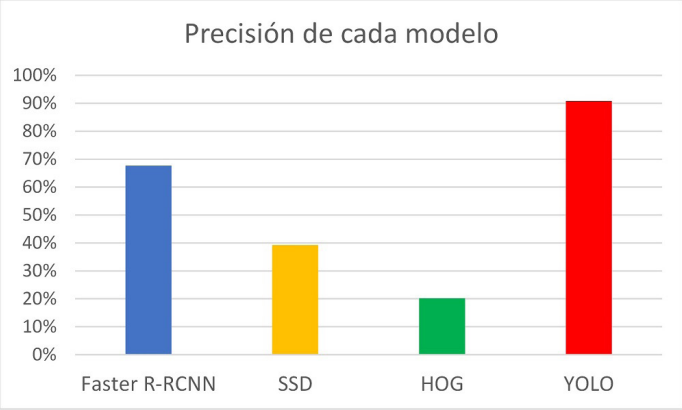

Figura. 3. Comparativa de precisión de cada modelo. (fuente propia)

En cuanto a la estimación de la exhaustividad (Figura 4), los resultados dejan a SSD como el modelo con el porcentaje más alto ya que este fue el modelo con más detecciones correctas realizadas. El porcentaje de Exhaustividad de este modelo fue de 95.71% seguido por YOLO con un porcentaje del 80.48% y Faster R-CNN con porcentaje del 82.85%. En este caso HOG vuelve hacer el modelo con el porcentaje más bajo siendo este de 30%.

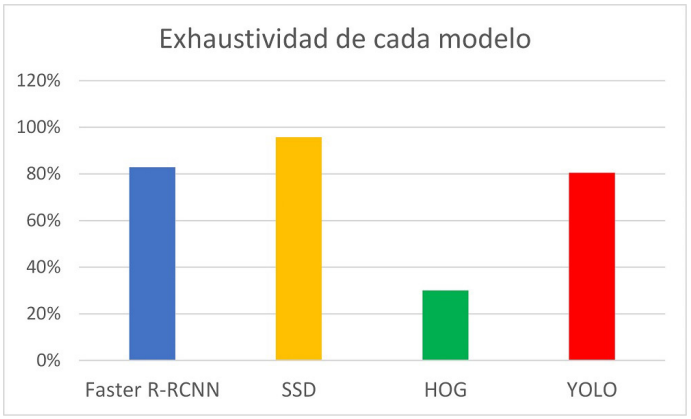

Figura. 4. Comparativa de Exhaustividad de cada modelo. Fuente: elaboración propia

A modo de resumen se puede resaltar que YOLO es el modelo más preciso, SSD es el que mejor exhaustividad tiene y HOG, pese a la velocidad demostrada en el apartado anterior, resulta ser el modelo menos preciso y la exhaustividad más baja esto debido a su bajo número de verdaderos positivos detectados y su alto número de omisiones al realizar la inferencia sobre las imágenes.

Resultados de error

Las métricas de error utilizadas en esta comparativa fueron, el error cuadrático medio, raíz del error cuadrático medio y error absoluto medio. En estas pruebas YOLO es el modelo que presenta menos errores y SSD es el modelo que presenta más errores en cuanto al conteo de personas se refiere. Los resultados de estas pruebas se pueden apreciar en la Tabla 9 y figura 6.

Tabla.10. Resultados de MSE, RMSE y MAE por modelo

Modelo	MSE	RMSE	MAE
Faster R-RCNN	1.0867	1.0424	0.6333
SSD	7.6533	2.7665	2.0400
HOG	2.3200	1.5235	1.1867
YOLO	0.2267	0.4761	0.2133

Fuente: Elaboración propia

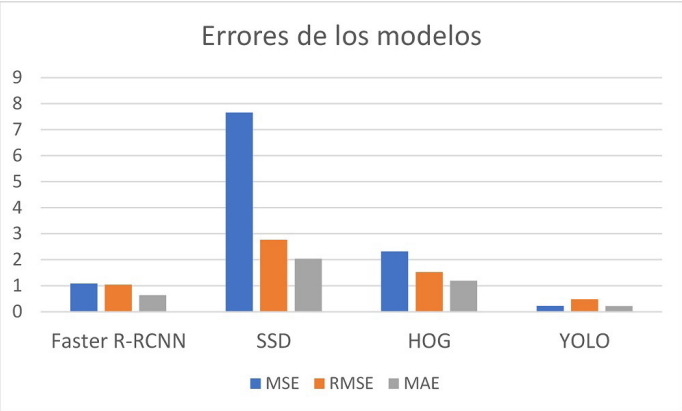

Figura. 5. Grafica de comparativa de errores de cada modelo.

Fuente: elaboración propia

Como se puede apreciar en la Figura 6, YOLO es el modelo que presenta los valores más bajos en con gran diferencia del resto mientras que SSD muestra un error cuadrático medio muy superior al resto de modelos, sin embargo, su error absoluto medio es muy cercano al de HOG.

CONCLUSIONES

En las pruebas de velocidad quedó demostrado que el modelo YOLO es el más rápido de los cuatro modelos puestos a prueba y por mucho, tanto en tiempo total de inferencia como en el tiempo medio de inferencia. Los resultados de este modelo indican que el tiempo medio de inferencia es de 0.0280 lo que implica que, en un segundo, el modelo es capaz de procesar 35 fps aproximadamente. Considerando que las cámaras de vigilancia pueden llegar a tener entre 15 a 30 fps, es factible considerar que el modelo puede trabajar en tiempo real.

De las pruebas de precisión se puede concluir que el modelo más preciso es YOLO ya que supera por bastante al resto de modelos, sin embargo, es de resaltar que YOLO no fue capaz de detectar todas las personas presentes en algunas imágenes (falsos negativos) y como este trabajo está pensado para ayudar en el control de aforo de establecimientos o ambientes cerrados de acceso público, para ayudar a prevenir los contagios de coronavirus aparte de una precisión alta también se requiere de una exhaustividad alta para asegurarse de detectar a todos los individuos presentes. En este caso, el modelo que tiene la mayor exhaustividad es SSD al haber realizado el mayor número de detecciones correctas con relación a las esperadas.

En cuanto a lo que se refiere únicamente a los errores en el conteo, es posible apreciar que YOLO es el modelo que menos errores presenta ya que en este caso únicamente se toma en cuenta la cantidad de personas detectadas contra las personas que realmente se encontraban en la imagen. En general se puede afirmar que el modelo con el mejor tiempo, con menos errores y con mayor precisión es YOLO, siendo únicamente desplazado en la exhaustividad por el modelo SSD.

Este trabajo fue realizado, para tenerlo en cuenta como un punto de partida al momento de querer implementar un sistema detección de personas para el control de aforo en ambientes cerrados que utilizan cámaras de vigilancia y modelos de inteligencia artificial de detección de objetos pensado principalmente para controlar el aforo en ambientes interiores y de esta forma ayudar a prevenir el contagio del COVID-19. Como posibles trabajos futuros se propone, la realización de comparativas con otros modelos o implementaciones que no se tomaron en cuenta en este trabajo para tratar de salvar la brecha que existe entre la precisión y la exhaustividad. Así como evaluar otros factores como la posición y cantidad de cámaras o las oclusiones entre personas.

También se propone trabajar en el entrenamiento de modelos con datasets más específicos ya que MS COCO es un conjunto de datos bastante general e incluye diferentes clases que para fines de esta investigación no eran tan relevantes. Para esto podría realizarse una transferencia de aprendizaje (transfer learning) para aprovechar el entrenamiento de los modelos existentes y de esta forma solo entrenar para detectar personas en los ambientes deseados.

REFERENCIAS BIBLIOGRAFICAS

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. Geosci. Model Dev. (Geoscientific Model Development), 7(3), 1247–1250. doi:10.5194/gmd-7-1247-2014

Ciotti, M., Ciccozzi, M., Terrinoni, A., Jiang, W.-C., Wang, C.-B., & Bernardini, S. (2020). The COVID-19 pandemic. Critical Reviews in Clinical Laboratory Sciences, 57(6), 365–388. doi:10.1080/10408363.2020.17831988

Gupta, P., Sharma, V., & Varma, S. (2021). People detection and counting using YOLOv3 and SSD models. Materials Today: Proceedings. doi: 10.1016/j.matpr.2020.11.562

Huang, T. (1996). Computer Vision: Evolution and Promise. 19th CERN School of Computing, (págs. 21-25). doi:10.5170/CERN-1996-008.21

Kumar Singh, A., Singh, D., & Goyal, M. (2021). People Counting System Using Python. 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 1750–1754. doi:10.1109/iccmc51019.2021.9418290

Li, J., Yin, Y., Liu, X., Xu, D., & Gu, Q. (2017). 12,000-fps Multi-object detection using HOG descriptor and SVM classifier. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (págs. 5928–5933). doi:10.1109/IROS.2017.8206487

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., . . . Dollár, P. (2014). Microsoft COCO: Common Objects in Context. https://arxiv.org/abs/1405.0312v3

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. European Conference on Computer Vision, 9905, págs. 21-37. Cham. doi:10.1007/978-3-319-46448-0_2

Marroquin, R., Dubois, J., & Nicolle, C. (2019). WiseNET: An indoor multi-camera multi-space dataset with contextual information and annotations for people detection and tracking. Data in Brief, 27. doi: 10.1016/j.dib.2019.104654

McCarthy, J. (2007). What is artificial intelligence? http://35.238.111.86:8080/jspui/bitstream/123456789/274/1/McCarthy_John_What%20is%20artificial%20intelligence.pdf

Padilla, R., Netto, S. L., & da Silva, E. A. (2020). A Survey on Performance Metrics for Object-Detection Algorithms. International Conference on Systems, 237–242. doi:10.1109/iwssip48289.2020.9145130

Pisner, D. A., & Schnyer, D. M. (2019). Chapter 6 - Support vector machine. En A. Mechelli, Machine learning (págs. 101–121). San Deigo: Elsevier. doi:10.1016/B978-0-12-815739-8.00006-7

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. doi:10.1109/cvpr.2016.91

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137–1149. doi:10.1109/tpami.2016.2577031

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 658-666. doi:10.1109/cvpr.2019.00075

Rich, E. (1985). Artificial intelligence and the humanities. Computers and the Humanities, 19(2), 117–122. http://www.jstor.org/stable/30204398

Sarria-Guzmán, Y., Fusaro, C., Bernal, J. E., Mosso-González, C., González-Jiménez, F. E., & Serrano-Silva, N. (2021). Knowledge, Attitude and Practices (KAP) towards COVID-19 pandemic in America: A preliminary systematic review. J Infect Dev Ctries (The Journal of Infection in Developing Countries), 15(1), 9–21. doi:10.3855/jidc.14388

Shukla, R., Mahapatra, A. K., & Peter, J. S. (2021). Social distancing tracker using yolo v5. Turkish Journal of Physiotherapy and Rehabilitation, 32(2), 1785-1793.

Zou, Z., Shi, Z., Guo, Y., & Ye, J. (2019). Object Detection in 20 Years: A Survey. https://arxiv.org/abs/1905.05055