



POSGRADOS

MAESTRÍA EN _____ ELECTRÓNICA Y AUTOMATIZACIÓN

RPC-SO-19-No.277-2018

OPCIÓN DE
TITULACIÓN:

ARTÍCULOS PROFESIONALES DE ALTO NIVEL

TEMA:

DESARROLLO DE UN SISTEMA DE DETECCIÓN DE
DISTANCIAMIENTO SOCIAL MEDIANTE ALGORITMOS DE
INTELIGENCIA ARTIFICIAL

AUTOR:

DARWIN ALEXANDER MERIZALDE JIMENEZ

DIRECTOR:

PAULINA ADRIANA MORILLO ALCÍVAR

QUITO - ECUADOR
2021

Autor/a:



Darwin Alexander Merizalde Jiménez
Ingeniero en Electrónica en Automatización y Control
Candidato a Magíster en Electrónica y Automatización con
Mención en Informática Industrial por la Universidad Politécnica
Salesiana – Sede Quito.
dmerizaldej@est.ups.edu.ec

Dirigido por:



Paulina Adriana Morillo Alcívar
Master Universitario en Gestión de la Información
Ingeniera en Electrónica y Telecomunicaciones
pmorillo@ups.edu.ec

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

DERECHOS RESERVADOS

©2021 Universidad Politécnica Salesiana.

QUITO – ECUADOR – SUDAMÉRICA

MERIZALDE JIMENEZ DARWIN ALEXANDER

***DESARROLLO DE UN SISTEMA DE DETECCIÓN DE DISTANCIAMIENTO
SOCIAL MEDIANTE ALGORITMOS DE INTELIGENCIA ARTIFICIAL***

Desarrollo de un sistema de detección de distanciamiento social mediante algoritmos de inteligencia artificial.

Darwin Alexander Merizalde Jiménez and Paulina Adriana Morillo Alcívar

Universidad Politécnica Salesiana - Ecuador

Resumen

La pandemia actual del COVID-19 ha traído, sin duda, nuevos desafíos a la sociedad y la búsqueda constante de soluciones para controlar y disminuir sus efectos. En este sentido, el uso de la tecnología ha sido fundamental para sobrellevar la situación. De esta forma, este trabajo propone la implementación de un Sistema de Detección de Distanciamiento Social automático mediante algoritmos de Inteligencia Artificial y el uso de un vehículo aéreo no tripulado. Este sistema consta de dos bloques fundamentales, el primero que consiste en la creación y entrenamiento de una red neuronal convolucional para la detección de personas, utilizando el sistema de detección de objetos YOLO y el segundo que consiste en la adquisición y análisis del vídeo en tiempo real, esto implica la detección de personas y el cálculo de las distancias entre ellas. De modo que, si se incumple el distanciamiento social sugerido se puedan emitir alertas que faciliten la toma de decisiones. Esta solución es capaz de incrementar las capacidades de observación y vigilancia ciudadana para ayudar a las autoridades y a las fuerzas de seguridad a controlar posibles brotes de contagios masivos por COVID-19. Los experimentos se realizaron en tres escenarios distintos de vuelo a 15, 30 y 50 metros, respectivamente. De acuerdo a los resultados, la sensibilidad del sistema para la detección de personas alcanza valores superiores al 95%, aunque, los valores más altos se obtienen en vuelos a 30m de altura. En cuanto al cálculo de las distancias, en los tres escenarios el error relativo promedio es menor al 5%. Finalmente, se evalúan los tiempos de latencia de la transmisión del vídeo que en general son tiempos menores a 5 segundos. Por tanto, se puede concluir que el sistema devuelve resultados confiables para controlar el cumplimiento de medidas como el distanciamiento social.

Abstract

In this research work, the implementation of a Social Distancing Detection System is proposed through the use of Artificial Intelligence algorithms from any type of UAV. This system consists of two fundamental phases, the first is the person detection part, for which it is necessary to use the YOLO object detection system, based on the use of Convolutional Neural Networks whose networks are the most efficient working with images because they detect characteristics that determine the desired object according to the established learning and the rigorous training process applied to the Neural Network. The second phase consists of the video analytics that consists of the Communication with the UAV module, which deals with the forms and methods used for the computer to communicate with the UAV and the acquisition of video for its process and the second module the part of detection of the distancing with the respective alert, which is practically the estimation of the distances between the people recognized by the neural network in order to know exactly and precisely the violations of social distancing by means of the geometric calculation to obtain an estimate

of the distance seen from the drone camera lens. This solution is capable of increasing citizen observation and surveillance capabilities to help authorities and security forces control these massive COVID-19 infections.

Palabras clave— Drones; Aglomeraciones; COVID-19; Redes Neuronales Convolucionales; Visión Artificial; Distanciamiento; Protocolos de Comunicación, UAV .

1 Introducción

En la actualidad la sociedad afronta uno de los mayores retos provocados por el COVID 19 (Gawde, 2020), la necesidad de adaptarse a una nueva realidad, que a pesar del avance de la vacunación, sigue exigiendo el uso de mascarillas y distanciamiento social. Toda la sociedad y las ramas de la ciencia se han visto avocadas a buscar soluciones que permitan controlar y mitigar los efectos de la pandemia. De esta forma, en el campo de la inteligencia artificial (IA) han cobrado protagonismo los algoritmos de detección, seguimiento, reconocimiento, localización y conteo de personas (Chan et al., 2008), con la finalidad de controlar las aglomeraciones, detectar focos de contagio y ofrecer seguridad sanitaria a las personas.

De igual forma, según (Naqiyuddin et al., 2020) es necesario emplear herramientas tecnológicas para el control de los contagios masivos de COVID-19 y evitar la propagación del virus. Las acciones como el distanciamiento social y la disminución de las aglomeraciones son esenciales para este propósito, pero en muchos escenarios estas medidas son difíciles de controlar, por lo que se debe contar con nuevas herramientas que permitan verificar su cumplimiento. Los algoritmos de IA son muy eficaces para esta tarea, sin embargo, no trabajan solos, estos van de la mano con los sistemas de video-vigilancia, los cuales permiten obtener las imágenes de las regiones que se requieren controlar. Su funcionamiento está basado en el uso de cámaras, instaladas en puntos fijos o cámaras integradas en dispositivos móviles (Rashied Hussein et al., 2020), los segundos facilitan el monitoreo, especialmente, en zonas de difícil acceso o alto riesgo. En este sentido, dispositivos tecnológicos como los drones o vehículos aéreos no tripulados (UAV, por sus siglas en inglés), han resultado ser muy útiles en el monitoreo de la crisis sanitaria provocada por el COVID 19, logrando cubrir grandes áreas y determinando focos de infección (Kumar et al., 2021). Su integración con algoritmos de inteligencia artificial han permitido realizar el monitoreo de la población de manera casi automatizada (Degadwala et al., 2020), facilitando el control de las medidas de bioseguridad. A esto se suma, el hecho de que, la mayoría de los sistemas actuales de vídeo-vigilancia actúan de manera activa, a través de los sensores ópticos, que les da la capacidad de identificar señales, segmentarlas, separarlas, reconocerlas e interpretar una escena, en la mayoría de los casos en tiempo real, gracias al desarrollo de la visión por computadora y las comunicaciones inalámbricas (Siriwardhana et al., 2020).

El desarrollo de tecnologías en los últimos años ha generado grandes avances especialmente en aquellos relacionados al campo de la ingeniería, la visión artificial es una herramienta que permite solucionar una serie de inconvenientes complejos para el ser humano y que mediante su aplicación pueden llegar a facilitar y mejorar la detección de rostros, distancias y una serie de patrones. Según (Outay et al., 2020) se hace necesario el empleo de las herramientas tecnológicas para el control de los contagios masivos de COVID-19, evitando así la propagación del virus.

En el caso del uso de UAVs, la incorporación de cámaras a vehículo aéreos no tripulados (UAVs) ha facilitado aun más el monitoreo y provocado un gran impacto en el tema de vigilancia de áreas (Chamola et al., 2020), ya que estos artefactos pueden alcanzar zonas geográficas de difícil acceso y hacer el seguimiento de grandes regiones de forma muy eficiente (Rashied Hussein et al., 2020).

Trabajos recientes muestran la efectividad de algoritmos de visión artificial para determinar: lugares con aglomeraciones de personas, la temperatura corporal de una persona, el uso de mascarillas, distanciamiento social, entre otros. Por ejemplo, en (Bouhleb et al., 2021) se muestra el uso de redes neuronales convolucionales (CNN) pre-entrenadas como AlexNet y videos capturados por drones, para determinar zonas de alta densidad de personas y medir el distanciamiento social. El algoritmo que implementan alcanza para la primera fase de clasificación más del 80% de exactitud y cerca del 100% para el cálculo de las distancias. De igual forma, en el trabajo de (Barnawi et al., 2021) se usan UAVs equipados con cámaras y sensores térmicos para capturar información relacionada con la temperatura corporal y determinar posibles casos de COVID. Así mismo, presentan un esquema de detección de rostros y uso de la mascarilla, obteniendo exactitudes mayores al 99%. En otro trabajo (Somaldo et al., 2020) se realiza una implementación que implica la detección de personas, también mediante el uso de redes neuronales convolucionales en este caso YOLO (*You Only Look Once*), el trazado de distanciamiento entre las personas del escenario, con la detección de quienes están violando el distanciamiento social, se realiza sobre entornos virtuales, utilizando las plataformas *Gazebo* y *Robotic Operative System* (ROS), cuyas funciones facilitan la virtualización de entornos reales en 3D.

Como se evidencia en la literatura, existe una variedad de trabajos, con distintos enfoques, que buscan mediante el uso de la tecnología y los algoritmos de inteligencia artificial mejorar el control de la pandemia. De igual forma, este artículo propone la implementación de un sistema de detección de distanciamiento social, pero en entornos reales externos, en escenarios captados mediante una cámara incorporada a un UAV, con un ángulo de efectividad de 0 a 45 grados. Para la identificación de los objetos de interés, en este caso las personas, se utilizan algoritmos de visión artificial y redes neuronales convolucionales pre-entrenadas como YOLO, luego de esta fase se realiza el cálculo de la distancia usando la geometría euclídeana en el espacio. Así, la metodología empleada se describe con mayor detalle en la Sección 2, los resultados experimentales de la implementación se exponen en la Sección 3 y finalmente, en la Sección 4, se presentan las conclusiones y prospectivas de este trabajo.

Según (Chan et al., 2008) hay una serie de trabajos enfocados en el conteo de personas y el distanciamiento social, el principio de estos trabajos está centrado en la extracción de imágenes o los cuadros de video. Para (Babu Sam et al., 2017) otros trabajos realizados utilizan la clasificación automática por medio de las *Convolutional Neural Networks (CNN)*, como una forma de clasificar píxeles por medio de la distribución Gaussiana y estableciendo una clasificación, así como niveles de entrenamiento. Es así como los modelos de regresión permiten obtener estimaciones a nivel global y local, enfocándose en la extracción de características multinivel en combinación con multisalida y modelos de regresión, esto da paso a la localización, reconocimiento de píxeles y el aprendizaje de características en un definitivo grupo de píxeles (Chen et al., 2021).

Según (Liu et al., 2018) los procesos genéricos son parte de los modelos predictivos, los mismos que están conformados por el procesamiento de la muestra, para más adelante someter a procesos de entrenamiento por medio de un set de entrenamiento y la validación del mismo y su comparación con el test set, con la finalidad que la predicción sea correcta. Para conseguir la estimación de personas en una zona cualquiera es por medio del cálculo convolucional de las características del objetivo (Srinivasan et al., 2021).

Para (Kang et al., 2019) los trabajos más importantes utilizan mapas de densidad, los cuales funcionan mediante un procesamiento donde se establecen regiones de píxeles los cuales cambian bruscamente de color en base a las características de los objetos lo cual permite identificar los puntos de mayor afluencia, en este caso de personas y por ende de mayor contagio. En (Lian et al., 2019) se instauran una serie de resultados mediante la toma de características de los píxeles utilizando RGB para la clasificación.

Según (Motlagh et al., 2017) otros trabajos realizados se enfocan en la vigilancia área de las aglomeraciones utilizando sistemas aéreos no tripulados. Según (Tzelepi and Tefas, 2017) señalan que los denominados UAVs utilizados para recabar información acompañados de algoritmos que identifican las personas que permiten una mejor localización del problema.

Para (Paredes et al., 2020) en trabajos anteriores se ha mejorado y verificado la utilización de drones para la vigilancia y reconocimiento de personas, de esta forma también se han utilizados mapas de densidad y la clasificación CNN para tener un estimado de número de personas en escenarios reales de monitoreo (Cruz et al., 2020). También según (Conte et al., 2021) se ha implementado mediante el uso de UAVs la detección de focos de contagio en las personas dentro un escenario específico mediante el uso de cámaras térmicas donde han visualizado y detectado la persona con mayor temperatura y así determinar quien puede tener COVID-19 en cualquier área, pero sin embargo esto requiere mucho presupuesto ya que las cámaras térmicas de larga distancia no son económicamente accesibles.

Mediante visión por computadora y el uso de CNNs para la detección de objetos. En este caso, se utiliza YOLO para detectar las personas y luego se estimará las distancias entre ellas, por medio de la geometría euclídeana en el espacio, tomando como referencias el ángulo de la cámara y la altura del UAV para transformar píxeles a unidades métricas y posteriormente presentar estos resultados en el dispositivo de visualización. Por lo tanto, en la siguiente sección se expone los dos fases indispensables para la implementación de este sistema. el primero que consiste en la detección de objetos mediante YOLO y la segunda fase consiste en la analítica de vídeo en tiempo real que incluye el módulo de comunicación con el UAV para la adquisición y procesamiento del vídeo, seguido el módulo de la estimación de distancias.

El aporte de este trabajo es una implementación de un detector de distanciamiento social capaz de emitir cualquier alarma para tomar acciones pertinentes al instante y momento oportuno en escenarios reales y con una metodología robusta y ligera como lo es el uso de YOLO, ya que en trabajos mencionados anteriormente lo implementan de forma virtual y con otras técnicas de estimación así como también el empleo de otras redes neuronales menos eficientes en cuanto a costo computacional. Los vuelos de prueba para este trabajo se han realizado a alturas próximas entre 15 hasta 50 metros siendo estas donde mejor se desempeña el estimador y el sistema de detección YOLO, esto dependerá también de la resolución de la cámara y las capacidades que tienen cada una de estas para hacer acercamientos a grandes alturas.

2 Metodología

La metodología llevada a cabo en la implementación del sistema tiene dos fases principales, tal y como se muestra en el diagrama de la Figura 1, la primera fase consiste en la gestión de la red neuronal convolucional utilizada para la detección de personas. Esta fase incluye la creación y entrenamiento del modelo y el etiquetado de las imágenes para el entrenamiento. Además comprende la evaluación y validación del modelo. La segunda fase es la gestión de la analítica de video que conlleva la comunicación con el dispositivo UAV, la adquisición y procesamiento del video en tiempo real, y la estimación de las distancias entre las personas. Los resultados del análisis se visualizan en una interfaz de usuario, que facilita el control y la toma de decisiones.

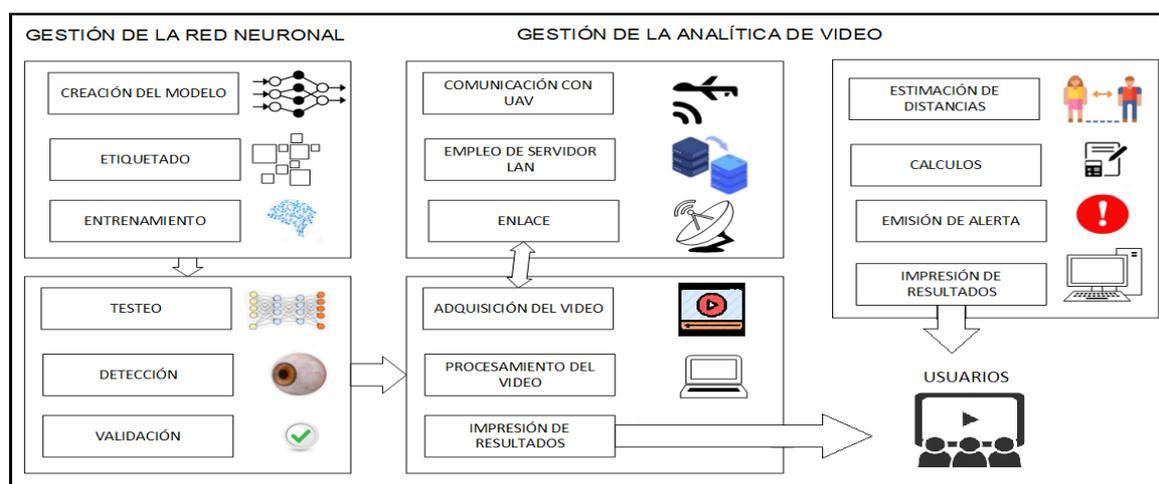


Figura 1: Esquema General de la Metodología de diseño

2.1 Configuración y entrenamiento de la red neuronal convolucional YOLO

La arquitectura de YOLO está basada en el modelo de clasificación de imágenes GoogleNet (Gupta et al., 2020). Esta CNN realiza la detección ajustando los píxeles de la imagen a una matriz de $S \times S$, si una clase u objeto a ser detectado se encuentra dentro de una de las cuadrículas generadas, esta se encarga de realizar la predicción asociando un valor de probabilidad.

La arquitectura de la red neuronal convolucional usada en este trabajo es la de YOLO optimizado y su estructura se muestra en el diagrama de la Figura 2, es decir la versión *fast* similar a la versión completa pero con un menor número de capas y filtros ya que consta de 1×1 capas de reducción y 3×3 capas convolucionales (Paredes et al., 2020) más las capas completamente conectadas, evitando así el uso de una red completa que emplea un total de 11×11 capas, lo que genera un alto costo computacional y a su vez encarece la implementación. (Madane and Chitre, 2021).

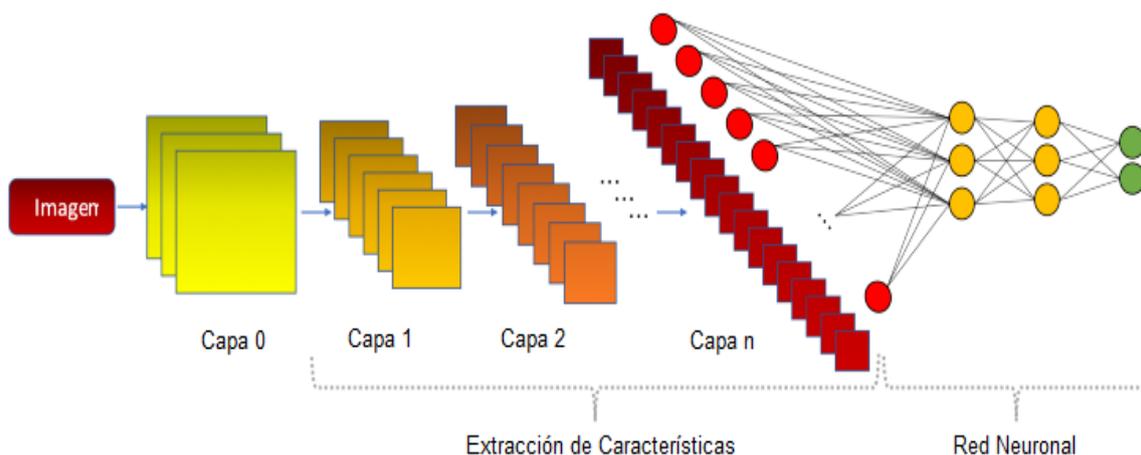


Figura 2: Arquitectura de una red neuronal convolucional para clasificación de imágenes (ICHI, 2021)

Toda la configuración de la red se lleva a cabo en el lenguaje multifuncional Python, el mismo que, gracias a sus características de programación intuitiva y que cuenta con el soporte adecuado en el uso de librerías para redes neuronales es ampliamente utilizado por muchos desarrolladores en aplicaciones de Deep Learning. El algoritmo es totalmente adaptativo y permite implementar cualquier tipo de modificaciones y cambios en el modelo matemático con el fin de mejorar la precisión del sistema. Para el desarrollo y funcionamiento de todo el sistema es indispensable conocer la manera de cómo se comunica cada uno de sus elementos y están estructurados como se muestra en el diagrama de la Figura 3

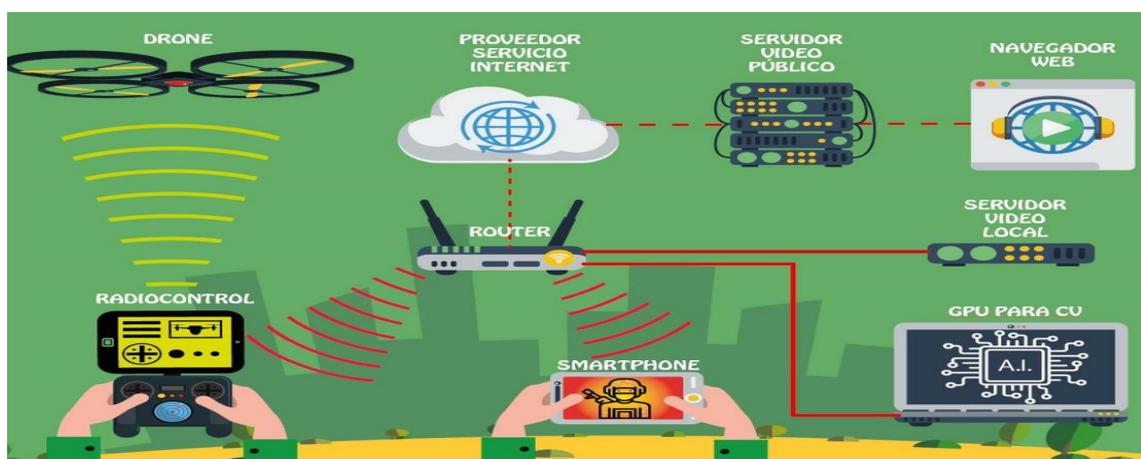


Figura 3: Estructura de hardware para la implementación (ICHI, 2021)

Para el entrenamiento de la red neuronal se requiere un conjunto de imágenes etiquetadas, en este caso se cuenta con una base de 1600 imágenes obtenidas de vuelos de UAVs con cámaras con definición 4k, captadas en vuelos a alturas desde 15 a 50 metros, con una cámara cuya resolución es de 14 mega píxeles y su field of view (FOV) es de hasta 45 grados, para mayor ángulo hasta perspectiva cenital es requerido otro entrenamiento (Shao et al., 2021).

El proceso de etiquetación de cada imagen consiste en realizar un trazado de forma manual de los objetos deseados en cada una de las imágenes, en este caso las personas, utilizando un software libre basado en Python denominado *LabelImg*, que guarda los puntos que conforman el rectángulo trazado (que representa al objeto) y lo posiciona en la matriz imagen exactamente donde se encuentra el objeto (Figura 4), de esta forma la CNN puede identificar exactamente la posición del objeto, para luego

clasificarlo. Posteriormente se genera un archivo *.xml* con anotaciones de la ubicación de cada clase y objeto dentro de la imagen (Ye et al., 2010), estos objetos son los cuadros de detección que contiene cada escenario indicando los objetos que debe aprender a detectar el algoritmo, como resultado de este proceso la máquina aprenderá a clasificar los objetos y marcar su posición en la escena con un cuadro de detección (Ye et al., 2010).



Figura 4: Proceso de etiquetado de las imágenes usadas en el entrenamiento de la red neuronal convolucional

Luego de realizar el etiquetado, se crea y entrena el modelo usando estas imágenes como entrada, la imagen de entrada es integrada con la matriz de etiquetado y luego es normalizada, dando como resultado una matriz unitaria que contiene la misma información, posteriormente estas imágenes pasan por las diferentes capas convolucionales, para extraer las características que permitirán su clasificación. El entrenamiento genera un modelo con la menor cantidad de pérdidas en la detección de los objetos. Una vez creado el modelo se procede a realizar un *test* con imágenes que no fueron parte del conjunto de entrenamiento, de esta forma se puede evaluar su rendimiento real. En caso de que el modelo no tenga un buen *performance* se lo puede afinar, depurando el *dataset* de entrenamiento o reconfigurando su número de capas. En el caso de este trabajo, la evaluación se realiza de forma manual en diferentes escenas de un video, por medio de la observación se detecta los items que han sido identificados o no por el algoritmo, esto permite calcular las métricas de rendimiento y determinar si el modelo está listo para ser utilizado en producción.

2.2 Análítica del video en tiempo real

En primer lugar, se realiza la adquisición del vídeo, a través de cámaras incorporadas en el UAV, previamente se inicializa ciertos parámetros del uso de hardware, como la activación de la GPU (*Graphical Process Unit*) y los valores de la altura y ángulo de cámara durante el vuelo, estas variables serán de vital importancia para el cálculo del distanciamiento. Una vez que se empiezan a capturar los *frames*, pasan a ser la entrada de la red neuronal convolucional (YOLO), que es la encargada de detectar los objetos (personas), en tiempo real, de la escena.

Luego de que las personas son identificadas en el vídeo y antes de calcular las distancias entre ellas, se realiza un conteo del número de objetos (n) detectados en la escena, de este modo se crea un matriz M de orden $[n \times 4]$, donde las columnas contienen las cuatro coordenadas del rectángulo que enmarca a cada objeto, después se calcula el centroide de cada rectángulo que servirá para determinar la distancia de ese objeto con respecto a todos los demás. Con los centroides de cada objeto se procede a calcular una matriz de distancias euclidianas ($D_{n \times n}$) donde cada elemento d_{ij} representa la distancia del objeto O_i al objeto O_j ($i = j = 1, 2, \dots, n$). Finalmente, cada distancia se compara con un valor umbral que de

acuerdo a las especificaciones de la OMS varía de 1,5 a 2m, de esta forma se verifica el cumplimiento del distanciamiento social, aquellas personas que no cumplen con esta separación se enmarcan en color rojo, para facilitar su monitoreo y emitir una alerta de violación a esta medida.

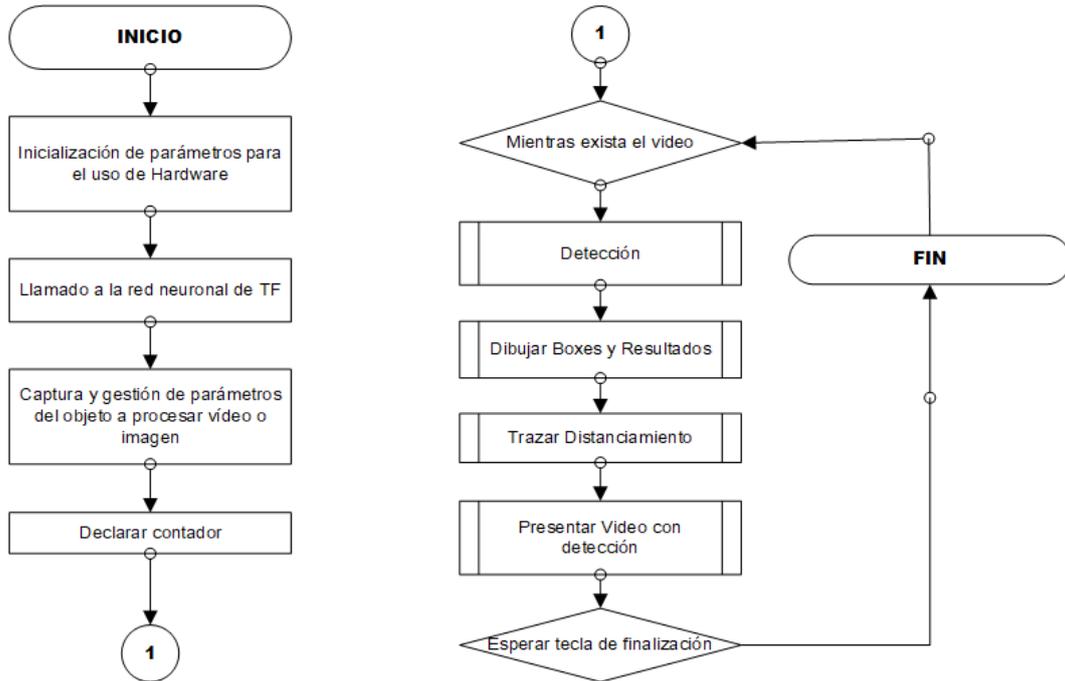


Figura 5: Diagrama de flujo del algoritmo de implementación.

En la Figura 5 se muestra todo el proceso, que empieza con la inicialización de parámetros del uso de hardware, como la activación de la GPU (*Graphical Process Unit*) y los valores de la altura y ángulo de cámara durante el vuelo, estas variables son vitales para el cálculo del distanciamiento. Posteriormente, se realiza el llamado de la red neuronal convolucional implementada para la detección de objetos (personas), y mediante la obtención del vídeo trabajar en cada fotograma que será el insumo de entrada durante el procesamiento, la frecuencia de muestreo del video es de 60 Hz. Al obtener el fotograma se declara una variable para contar los objetos detectados por YOLO, pasando a un bucle infinito que valida la existencia del fotograma, al no existir vídeo el sistema se hace recurrente hasta la detección de un fotograma. Una vez ingresado el fotograma, se aplica el proceso de detección mediante la CNN YOLO en el mismo que imprime los cuadros de detección, Para aplicar a la función del trazado de distanciamiento que realiza los cálculos de los centroides de cada rectángulo de detección las distancias entre centroides determina la traza las distanciamiento según los resultados obtenidos en el conteo, presentándolas también en los resultados, adicional a esto se presenta la existencia de violación al distanciamiento que está dado por la comparación de límite inferior de la estimación que en este caso sería dos metros, esto se presenta por medio de un mensaje de alerta en la interfaz del usuario, dando fin al proceso.

2.2.1 Comunicación con el UAV y adquisición del video

Para establecer la comunicación entre el dron y el sistema de analítica de vídeo se propone establecer los protocolos de transmisión de vídeo por medio de un servidor de *streaming* llamado *Real Time Multiple Message* (RTMP), este protocolo permite obtener el vídeo en tiempo real (Chalmers et al., 2021). Para ello hay que contar con un servidor de vídeo de tipo RTMP que permita la transmisión de vídeo a cualquier otro dispositivo El diseño se puede apreciar en la Figura 6.

De forma práctica, para lograr que el computador y el UAV puedan mantener comunicación entre sí,

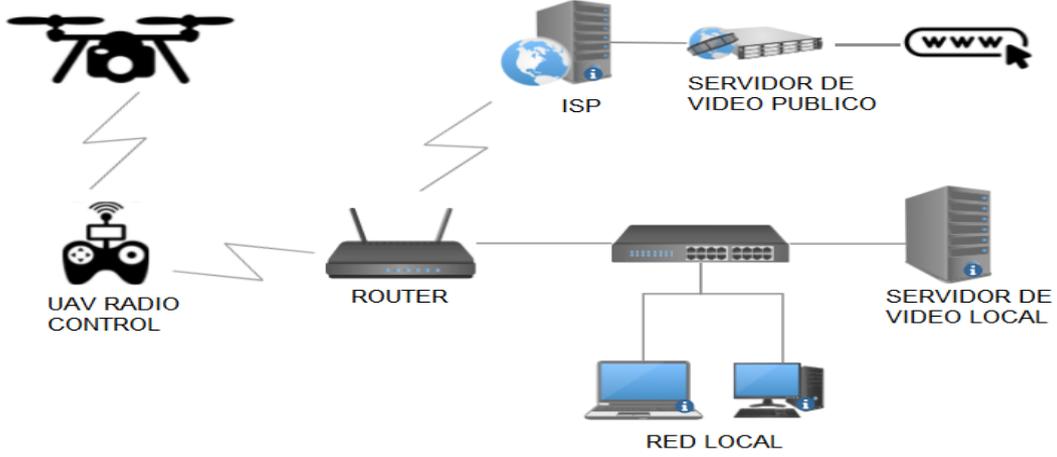


Figura 6: Esquema de comunicación del sistema.

se necesita crear una red interna con un servidor de vídeo intranet que permitirá capturar la visión del *UAV* de forma local desde cualquier computador y en la mejor resolución posible dependiendo del ancho de banda de la red y de la tasa de transmisión de vídeo, posterior a ello de forma opcional se puede emplear herramientas de transmisión de pantalla hacia la WAN como lo es OBS (*Open Broadcaster Studio*) en la que únicamente se debe redireccionar hacia el servidor de transmisión público y podrá ser visto con solo disponer la dirección del vídeo. Hay que tener en cuenta que solo es para visualización ya que la transmisión por WAN es simplemente la acción de compartir la pantalla donde se realiza el procesamiento hacia una herramienta que permite enviar *streaming* por medio de la WAN. (Doull et al., 2021).

2.2.2 Calculo del distanciamiento

Para obtener el espaciado, primero se obtienen los rectángulos que definen los objetos de detección, cada rectángulo se define con cuatro vértices con los que se calcula el centroide cite articuloDegadwala mediante la ecuación ref eq4. Esto localizará el centro de gravedad de la figura en cada detección.

Para la obtención del distanciamiento en este trabajo, primero se obtienen los rectángulos de detección, estos son contabilizados y cuentan con cuatro vértices en puntos específicos que conforman una la matriz del escenario y que define el objeto detectado por la CNN. Con estos rectángulos de detección se calcula el punto medio denominado centroide (Degadwala et al., 2020), este permitirá ubicar el centro de gravedad de la figura en cada detección. El centroide se puede hallar mediante la ecuación 4.

$$\bar{I}_x = \frac{bh^3}{12} \quad (1)$$

$$\bar{I}_y = \frac{b^3h}{12} \quad (2)$$

$$\bar{I}_{xy} = 0 \quad (3)$$

$$C = (\bar{I}_x, \bar{I}_y) \quad (4)$$

En donde \bar{I}_x representa el punto de inercia en el eje x , \bar{I}_y representa el punto de inercia en el eje de las y , \bar{I}_{xy} sería la inercia en ambos puntos que por lo general resultaría cero, b es valor de la base del rectángulo, h representa la altura del rectángulo de detección, para finalmente C representarlo como el centroide del rectángulo propiamente dicho, estas variables estarían dimensionadas en píxeles.

Una vez obtenidos los centroides o centros de gravedad de los objetos se toma como referencia el valor tanto para x y y de cada rectángulo de detección y se empieza a trazar una distancia euclidiana entre cada uno de los puntos de los centroides empleando la ecuación número 5, (Ahmad et al., 2020).

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (5)$$

En donde d representa la distancia que se desea obtener, x_1 es la posición del primer punto en x , x_2 es el valor del segundo punto en x , y_1 es la posición del primer punto en y , y_2 es el valor del segundo punto en y .

Para la realización del calibrado para la marcación de la distancia en base a la altura real y a la medida del alto del *frame* se realizó una calibración basada en semejanza de triángulos, tomando como referencia el ángulo de la cámara y la altura real del *UAV* mediante reglas pitagóricas se ha deducido la ecuación 6.

$$f_d = \frac{n}{h * \cos(\gamma - \alpha - \beta)} \quad (6)$$

Donde f_d representa el factor de ajuste de distancia que se aplicará al algoritmo para la estimación de la distancia dentro del escenario analizado, n representa el alto de la imagen en píxeles, h representa la altura a la que está ubicado el *UAV*, γ representa el ángulo máximo de vista de la cámara que es capaz de cubrir, α , representa el ángulo intermedio de referencia desde la posición de la cámara hasta el ángulo mínimo que la cámara es capaz de observar (Bhambani et al., 2020) y finalmente β es el ángulo muerto, que la cámara ya no es capaz de observar, todo esto sobre una perspectiva frontal con un ángulo de cámara superior a los 45 grados, ya que al ubicarse la cámara en una vista cenital el panorama cambia completamente siendo α igual a cero y el resto de ángulos distribuidos uniformemente con vista hacia ambos lados. Es importante mencionar que el sistema de detección *YOLO* mejora su rendimiento de reconocimiento de objetos con ángulos de cámara que van desde los 45 grados hasta los 90 grados (Aghaei et al., 2021). Todos estos cálculos están cargados dentro de la función de trazado de distanciamiento observado en la Figura 5.

Para evaluar el calculo de las distancias se utilizan dos métricas (Ecuación 7 y 8) que representan el error absoluto y el erro relativo, respectivamente. En ese caso, el término X_e es la medida estimada por el sistema, también llamado la distancia estimada y X_r representa la distancia real medida en el campo donde se realizaron las pruebas de estimación. Para el análisis de la distancia se realizarán las detecciones en tiempo real con el *UAV* y con la selección aleatoria de un fotograma desde dos escenarios diferentes a 15, 30 y 50 metros de altura, tomando así un conjunto de medidas que permitirán obtener una media aritmética del porcentaje de error de cada una de las distancias obtenidas tanto desde la cámara del dron y de las distancias reales del escenario que se encuentra en análisis. Para la métrica se van a poner en consideración las distancias marcadas por el sistema y se realizará una comparativa con las medidas tomadas en el escenario real. Para la obtención del error de medición será necesario el empleo del error absoluto expresada por la ecuación 7.

$$E_a = | X_e - X_r | \quad (7)$$

$$E_r = \frac{E_a}{X_e} = \frac{| X_e - X_r |}{X_r} \quad (8)$$

Para definir el error global del algoritmo empleado en el calculo de las distancias, se procederá a calcular la media de ambos errores E_a y E_r (Ecuación 9 y 10), sumando los errores cometidos en cada *frame* y dividiendo para el número total (m) de *frames* analizados.

$$\overline{E_a} = \frac{\sum_{i=1}^n | X_e - X_r |}{m} \quad (9)$$

Para la media del error relativo está disponible la ecuación 10.

$$\overline{E_r} = \frac{\sum_{i=1}^m \frac{E_a}{X_e}}{m} = \frac{\sum_{i=1}^m \frac{|X_e - X_r|}{X_r}}{m} \quad (10)$$

3 Experimentación y resultados

El bloque de detección de personas del sistema, que usa YOLO, está montado sobre *Tensorflow* y CUDA, mientras que para el tratamiento de vídeos e imágenes se utiliza *openCV*. En la Tabla 1 se muestran las especificaciones del hardware y software de la implementación del sistema.

Tabla 1: Especificaciones de Hardware y Software del Sistema.

Item	Especificación
Dron	Bebop-2 Parrot
Cámara	Nativa de Bebop 14 Mpxls
Computador	Laptop
Procesador	Intel Core i7
Memoria	16 GB RAM
GPU	NVIDIA, Geforce GTX-960
Sistema operativo	Windows 10
Versión Python	3.7
Versión OpenCV	4.1
Versión CUDA	10
Versión TensorFlow	1.5

En la fase de experimentación se han seleccionado de forma aleatoria diez fotogramas de tres escenarios diferentes, tomados a 15, 30 y 50 metros de altura (Escenario 1, Escenario2, Escenario 3, respectivamente). Las métricas consideradas en los experimentos fueron la sensibilidad del algoritmo en la detección de personas, el error en el calculo de las distancias y los tiempos de latencia en la transmisión del video.

Una de las principales métricas para analizar el sistema implementado será en la parte de detección en donde se analiza la sensibilidad que determinará la eficiencia en detección de un solo tipo de objeto en este caso las personas, dando a conocer la confiabilidad de la detección y que tan bien entrenada estuvo la red neuronal de detección de personas. Como otra parte del análisis de resultados se pondrá a prueba las estimaciones y se realizará una comparación con las medidas reales del evento que permiten obtener la eficiencia para detectar distancias.

Una vez implementado para el análisis de la distancia se realizarán las detecciones en tiempo real con el *UAV* y con la selección aleatoria de un fotograma desde dos escenarios diferentes a 15, 30 y 50 metros de altura, tomando así un conjunto de medidas que permitirán obtener una media aritmética del porcentaje de error de cada una de las distancias obtenidas tanto desde la cámara del dron y de las distancias reales del escenario que se encuentra en análisis. Para la métrica se van a poner en consideración las distancias marcadas por el sistema y se realizará una comparativa con las medidas tomadas en el escenario real. Para la obtención del error de medición será necesario el empleo del error absoluto expresada por la ecuación 7.

3.1 Sensibilidad del modelo en la detección de las personas

El calculo de la sensibilidad del modelo en la detección de personas se realizó usando la Ecuación 11 (Paredes et al., 2020).

$$sensibilidad = \frac{VP}{VP + FN} \quad (11)$$

Donde VP representa el número de objetos detectados que corresponden a personas y FN representa el número de personas que no han sido reconocidos en la escena. Los verdaderos positivos, que vendrían a ser los aciertos en los objetos entrenados que el sistema logró captar, encerrándolos en un cuadro y FN representa los falsos negativos que vienen a ser aquellos objetos que la red no logra detectar a pesar de estar presente y de haber sido entrenados, esto se debe a que la red se confunde al no percibir una cierta gama de colores o estos llegan a estar mimetizados con el entorno. Siendo todo esto una forma

de determinar si el sistema detecta o no a las personas en cada escenario, la clasificación de detecciones están bien definidas para que únicamente detecte personas.



Figura 7: Detección del fotograma con analítica de vídeo.

Tabla 2: Resultados Sensibilidad del modelo de detección en los tres escenarios

Fotogramas	Escenario 1 a 15m				Escenario 2 a 30m				Escenario 3 a 50m			
	Objetos	VP	FN	Sensibilidad	Objetos	VP	FN	Sensibilidad	Objetos	VP	FN	Sensibilidad
F1	20	18	2	0,9000	7	7	0	1,0000	6	6	0	1,0000
F2	21	20	2	0,9524	9	9	0	1,0000	2	2	0	1,0000
F3	22	20	2	0,9091	8	7	1	0,8750	8	6	2	0,7500
F4	19	16	3	0,8421	8	8	0	1,0000	6	5	1	0,8333
F5	21	20	1	0,9524	7	6	1	0,8571	6	6	0	1,0000
F6	22	19	3	0,8636	7	6	1	0,8571	8	7	1	0,8750
F7	19	18	1	0,9474	7	6	1	0,8571	5	5	0	1,0000
F8	20	18	2	0,9000	7	6	1	0,8571	6	5	1	0,8333
F9	22	20	2	0,9091	7	6	1	0,8571	3	2	1	0,6670
F10	17	16	1	0,9412	7	6	1	0,8571	5	5	0	1,0000
Media	20,3	18,5	1,8	0,9117	7,4	6,7	0,7	0,9018	5,5	4,9	0,6	0,8958

El fotograma de la Figura 7, tomado a 15m de altura y un ángulo de visión de 45 grados, muestra un ejemplo de los experimentos realizados, en este caso, se cuentan el número de rectángulos que hay en la escena y se realiza el cálculo de la sensibilidad, la Tabla 2 resume los resultados de la sensibilidad del modelo en la detección de personas. A pesar de, que la frecuencia del video es de 60Hz, para el procesamiento (detección de personas y cálculo de la distancia) únicamente se toman cuatro frames por segundo. Esto se realiza del mismo modo en los tres escenarios planteados.

Se puede observar que la sensibilidad es mayor en vuelos de 30 metros de altura, esto se debe a que la resolución de la cámara del UAV mejora a esa distancia ya que la red neuronal entrenada determina mejor las características del objeto y los reconoce como tal.

Con esta implementación se ahorra costo computacional evitando la latencia y la ralentización del vídeo cuando se aplica al tiempo real o como puede suceder la acumulación de fotogramas que el sistema no alcanza a procesar provocando un desbordamiento de la memoria del computador. Los resultados con datos numéricos tomados para la sensibilidad se ha generado una muestra aleatoria de 10 fotogramas desde una población de 531 fotogramas que contiene el vídeo cuyos resultados se muestran en la Tabla 2.

El entorno analizado está adecuado para la detección de personas, el ángulo y altura de la cámara se encuentra justamente para que el sistema esté en condiciones de obtener los resultados requeridos por lo que se ha proporcionado este vídeo con un correcto ajuste de zoom y de ángulo de visión de 45 grados, a esto se suma la resolución de la cámara que en este caso se trata de 14 mega píxeles de resolución ya que con ello permite la observación clara y oportuna de los objetos para lo que fue entrenada la red neuronal convolucional.

La red neuronal en cuanto a la detección de personas en cualquier entorno se puede decir que es capaz de detectar una persona o conjunto de personas con una eficiencia del 91,17% en su sensibilidad, lo que permite el empleo confiable de esta red neuronal con el entrenamiento implementado.

3.2 Estimación del error en el calculo de la distancia

Las distancias medidas en los fotogramas de los diferentes escenarios se contrastan con la medición real de la distancia medido manualmente a través de un odómetro mecánico que ha permitido tomar el dato exacto real desde un individuo hacia otro, permitiendo calcular el error cometido en la estimación realizada por la CNN.

Las pruebas se realizaron a diferentes alturas determinadas, 15m, 30m y 50m para la evaluación de la precisión de las estimaciones de distanciamiento, en donde será posible observar la exposición de resultados obtenidos tanto por el sistema como en el campo propiamente dicho, esto se realizó gracias a un odómetro que permitió la obtención de una distancia exacta del escenario en el que se desarrollaron las pruebas. La estimación lo realizó el sistema de cálculo propuesto en este trabajo y utilizando estos dos valores se logró obtener los errores tanto absolutos y errores relativos, en base al error relativo se obtuvo la precisión del sistema para detectar distancias, y posterior a ello se calcularon los promedios respectivos.

Se muestra el sistema a continuación se presentan las imágenes con los resultados de la estimación en los escenarios ya antes mencionados el primero se realizará a una altura de 15 metros y se verifica cómo el sistema realiza su dimensionamiento, calcula y hace mediciones de distancia y las presenta en el vídeo en tiempo real, obteniendo así los datos para poder realizar la evaluación del sistema, los datos de la estimación de distancias son anotadas en una tabla de datos, las métricas se presentan en la Figura 8:

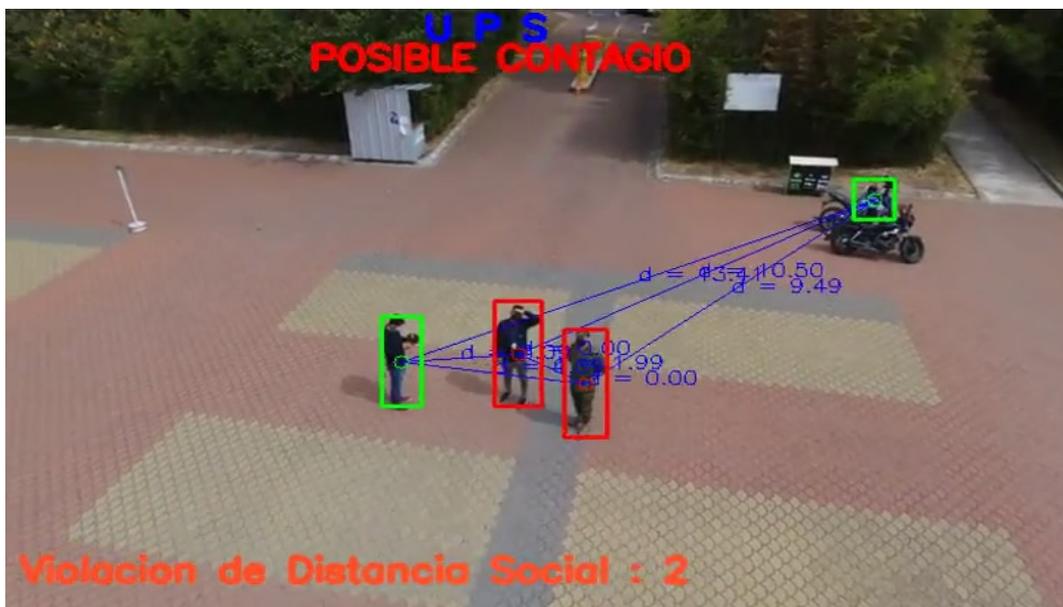


Figura 8: Fotograma para la estimación de distancias a 15 metros.

Los resultados y los errores de medición se detallan en la tabla 3 a una altura se 15 metros, en donde sus datos están tabuladas las distancias reales con las estimadas, obteniendo así un valor de precisión

aceptable.

Tabla 3: Resultados del distanciamiento a 15 metros

ORD	Dist. Medida	Dist. Real	E_a	E_r	Precisión %
1	13,41	10,38	3,03	0,29	70,81 %
2	10,5	8,79	1,71	0,19	80,55 %
3	9,49	9,21	0,28	0,03	96,96 %
4	1,99	1,50	0,48	0,33	67,33 %
5	2,89	2,16	0,73	0,34	66,20 %
6	2,09	2,55	0,54	0,35	65,16 %
Media			1,13	0,25	74,50 %

A continuación se presentan las imágenes con los resultados de la estimación en el segundo escenario a una altura de 30 metros, en esta altitud se debería mostrar datos mas cercanos a la realidad, sin embargo los datos de la estimación de distancias son anotadas en una nueva tabla de datos, las métricas se presentan en la Figura 9:

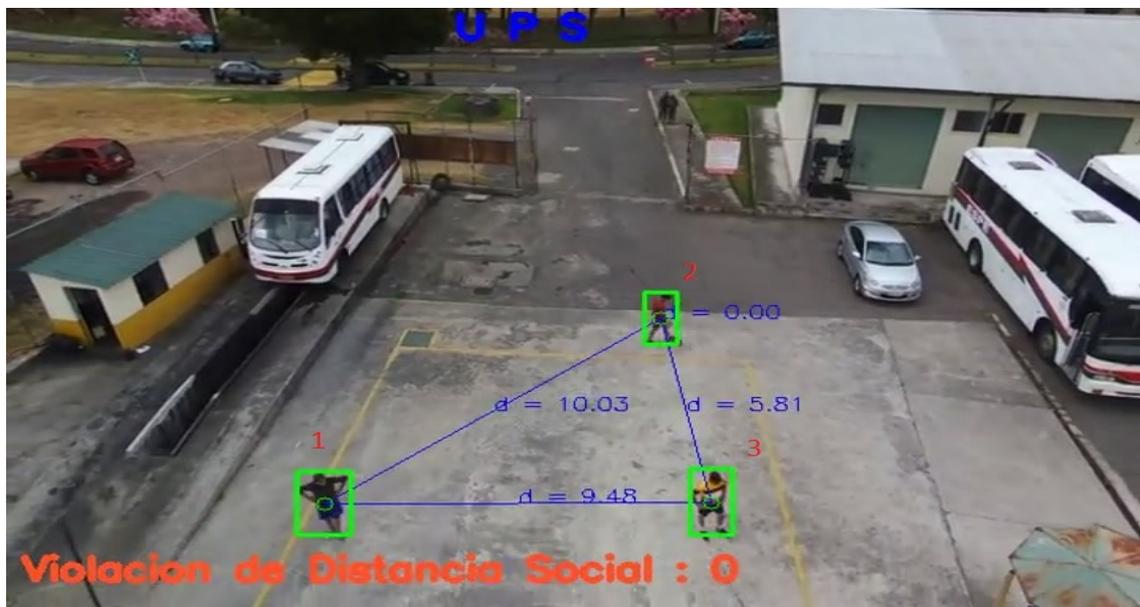


Figura 9: Fotograma para la estimación de distancias a 30 metros.

Los resultados y los errores de medición se detallan en la tabla 4 en donde sus datos están tabuladas las distancias reales con las estimadas, obteniendo así un valor de precisión mucho mejor que a 15 metros.

Tabla 4: Resultados del distanciamiento a 30 metros

ORD	d	Dist. Medida	Dist. Real	E_a	E_r	Precisión %
1	1-2	10,03	10,35	0,32	0,03	97 %
2	2-3	5,81	6,12	0,31	0,05	95 %
3	1-3	9,48	9,32	0,16	0,02	98 %
Media				0,26	0,03	96,67 %

A continuación se presentan las imágenes con los resultados de la estimación en un tercer escenario a una altura de 50 metros, a esta altitud se observa resultados aceptables pero ya alejados de la realidad,

los datos de esta estimación son anotadas en una nueva tabla de datos, las métricas se presentan en la Figura 10:

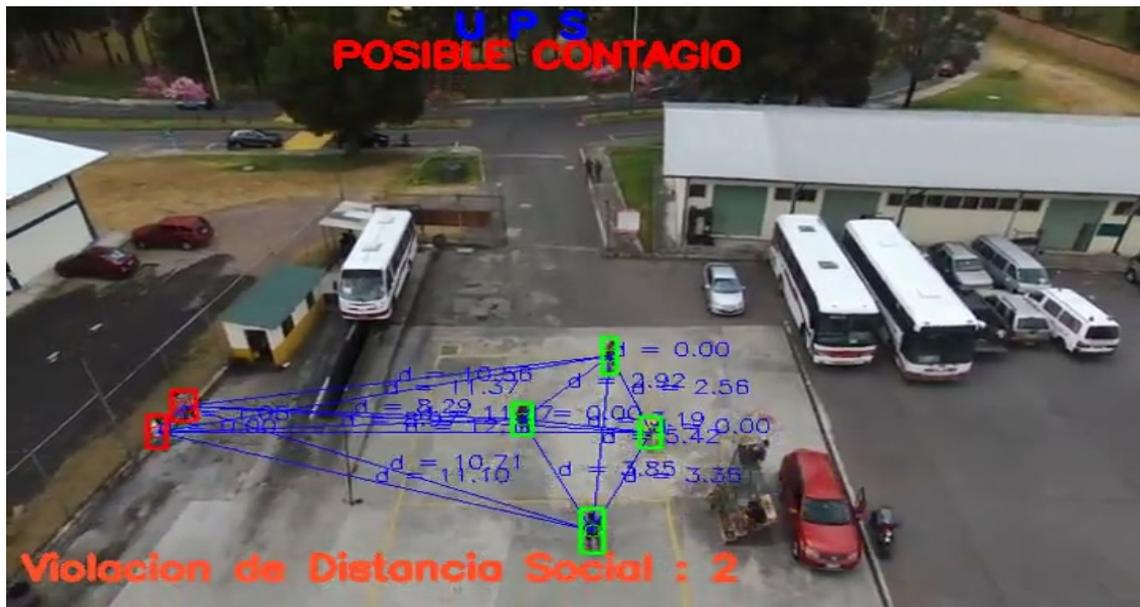


Figura 10: Fotograma para la estimación de distancias a 50 metros.

Los resultados de las estimaciones y errores aparecen en la tabla 7 en donde los datos están tabuladas las distancias reales y las estimadas, obteniendo así un valor de precisión menor que a 30 metros pero mejor que a 15 metros:

Tabla 5: Resultados del distanciamiento a 50 metros

ORD	Dist. Medida	Dist. Real	E_a	E_r	Precisión %
1	1,05	1,21	0,16	0,13	86,78 %
2	10,56	12,25	1,69	0,14	86,20 %
3	11,37	12,86	1,49	0,12	88,41 %
4	8,29	9,75	1,46	0,315	85,03 %
5	8,97	9,80	0,83	0,08	91,53 %
6	10,71	10,00	0,71	0,07	92,90 %
7	11,10	10,00	1,10	0,11	89,00 %
8	11,17	12,00	0,83	0,07	93,08 %
9	12,02	13,50	1,48	0,11	89,04 %
10	2,92	3,10	0,18	0,06	94,19 %
11	2,56	3,00	0,44	0,15	85,33 %
12	3,19	3,50	0,31	0,09	91,14 %
13	5,42	4,85	0,57	0,12	88,25 %
14	3,85	3,25	0,60	0,18	81,54 %
15	3,353	3,00	0,35	0,12	88,33 %
Media			0,81	0,11	88,72 %

Las Figuras 8, 9, 10 muestran un claro ejemplo de las distancias entre las personas calculada por el sistema en cada escenario, es decir, a diferentes alturas. En cada fotograma se muestran las distancias con líneas de color azul, los valores que calcula el sistema se comparan con los respectivos valores reales, de esta forma se calcula el error absoluto y relativo cometido en cada distancia que aparece en la escena

de la imagen (Tablas 3, 4, 7). Finalmente, se calcula el error promedio cometido en cada fotograma y este proceso se repite en los otros dos escenarios. Los resultados y los errores promedios de cada fotograma se resumen en la tabla 6.

Tabla 6: Resumen de las pruebas realizadas en la estimación de distancias

Fotogramas	Escenario 1			Escenario 2			Escenario 3		
	n	$\bar{E}a$	$\bar{E}r$	n	$\bar{E}a$	$\bar{E}r$	n	$\bar{E}a$	$\bar{E}r$
F1	20	0,1700	0,0170	7	0.1800	0.0178	6	0,3100	0,0163
F2	21	0,0500	0,0250	9	0.3200	0.0321	6	0,0500	0,0250
F3	22	0,0500	0,0250	8	0.3100	0.0312	6	0,3900	0,0780
F4	19	0,0800	0,0160	8	0.2400	0.0243	6	0,4300	0,0253
F5	21	0,2600	0,0200	7	0.2900	0.0289	6	0,3200	0,0363
F6	22	0,1300	0,0433	7	0.1800	0.0178	6	0,3500	0,0463
F7	19	0,2500	0,0500	7	0.2300	0.0229	6	0,4500	0,0023
F8	20	0,1800	0,0200	7	0.3100	0.0311	6	0,2000	0,0133
F9	22	0,3100	0,3100	7	0.2900	0.0288	6	0,3100	0,0155
F10	17	0,1400	0,0467	7	0.3200	0.0317	6	0,1100	0,0069
Media	20,3	0,1620	0,0573	7,4	0,2670	0,0267	5,5	0,2920	0,0265

En el primer escenario, el caso de altura a 15 metros se obtuvo un error absoluto promedio 0,1620 metros de error por cada medición, un error relativo promedio de 0,0573 que a su vez equivale al 5,7 % de error por cada medida, deduciendo una precisión del 94 % aplicando el sistema a esta altura.

En el segundo caso con altura a 30 metros se obtuvo un error absoluto promedio 0,2670 metros de error por cada medición, un error relativo promedio de 0,0267 que a su vez equivale al 2,67 % de error por cada medida, deduciendo una precisión del 97,33 % aplicando el sistema a esta altura.

Finalmente en el tercer caso con altura a 50 metros se obtuvo un error absoluto promedio 0.2920 metros de error por cada medición que es equivalente a un error relativo promedio de 0,0265 que a su vez equivale al 2,65 % de error por cada medida, deduciendo una precisión del 97,35 % aplicando el sistema a esta altura.

3.3 Rendimiento en la transmisión del vídeo

Un problema particular dentro de un sistema que funciona en tiempo real es la latencia de transmisión del vídeo, ya que de esto depende el resto de procesos como la detección, calculo de la distancia y el tiempo de retardo del vídeo obtenido en relación al tiempo presente, produciendo pérdidas de fotogramas y afectando al rendimiento del vídeo.

Las pruebas de comunicación con el *UAV* se han realizado desde un dron *Bebop 2* de la marca *PARROT* con una aplicación móvil llamada *Larix Screencaster* que permite la comunicación con cualquier tipo de servidor de vídeo RTMP. Este protocolo permite transmitir el video en *Streaming*, en el caso de *UAV* mas robustos como los *DJI* se puede transmitir mediante sus aplicaciones nativas. Se han realizado pruebas de conexión por este método durante el período de pruebas de la cual se han logrado mas de 100 conexiones de transmisión de vídeo y ninguna ha salido con falla alguna, permitiendo determinar que el sistema de comunicación no tiene problemas de efectividad ya que el servidor es quien realiza los cálculos para configurar la frecuencia de transmisión mas adecuada para evitar latencias durante la transmisión. En la tabla 7 se puede observar cómo cambia el rendimiento del vídeo capturado en función de la velocidad de la red disponible, por ejemplo si se obtiene un rendimiento del 50 % significa que el servidor dejará pasar uno por cada dos fotogramas, es decir que se perderán la mitad de fotogramas que envía el *UAV* o el dispositivo de captura de vídeo. Estos análisis se han realizado con

una resolución de cámara de 740x480 Mega píxeles.

Tabla 7: Análisis de rendimiento en transmisión de vídeo para todos los escenarios

Frecuencia	Velocidad de transmisión	Velocidad de Red	Fps	Rendimiento (%)
60 Hz	20 Mbps	10 Mbps	30	50
30 Hz	10 Mbps	10 Mbps	30	100
15 Hz	5 Mbps	10 Mbps	60	200

En Referencia a la frecuencia de transmisión del sistema de detección, el servidor proporciona 30 a 25 fotogramas por segundo, obteniendo así una transmisión natural y en tiempo real, pero el procesamiento de la red neuronal, la detección y el cálculo de las distancias requiere un tiempo prudencial que es añadido al tiempo de transmisión y al transcurrir miles de fotogramas estas fracciones de tiempo aumentan de forma divergente que podría causar retardos y latencia de transmisión. Para solucionar este inconveniente se configuró para que se haga la detección cada 10 fotogramas, así se da tiempo suficiente para que el sistema realice los cálculos y los resultados se sincronicen adecuadamente en la pantalla.

4 Conclusiones y prospectivas

Este trabajo presentó un sistema de detección de distanciamiento social en tiempo real. Según los resultados, el Recall del modelo de detección de personas se acercó al 90%. Por otro lado, los errores en la estimación de las distancias fueron inferiores al 3%, lo que significa que el sistema es útil para emitir alertas a las personas que están violando el distanciamiento con el fin de evitar posibles nuevas infecciones por COVID-19.

Durante el desarrollo de este trabajo se entrenó una Red Neuronal Convoucional que ha permitido mejorar la detección de objetos desde diferentes perspectivas desde los UAV con el fin de realizar un análisis desde cualquier escenario en tiempo real.

De acuerdo a los resultados obtenidos en la experimentación, la precisión de modelo es muy alta y se adapta a diferentes escenarios. Este trabajo resulta de alta validez para la sociedad ya que permite la detección del distanciamiento de las personas mediante el uso de inteligencia y visión artificial por computadora.

El sistema puede tener múltiples alertas, ya que se puede configurar para que las alertas se den por medio de sonidos u otros recursos disponibles, a fin de poder recomendar al personal y evitar contagios masivos.

Según las pruebas de vuelo realizadas a diferentes alturas 15m, 30m y 50m se obtuvieron precisiones de 74,5%, 96,67% y 88,72%, respectivamente, pudiendo determinar que la altura que logró mayor precisión fue de 30m, esta longitud es adecuada para propósitos de control de distanciamiento social. Sin embargo, con el resto de alturas se puede trabajar con ciertos ajustes. Hay que considerar también que en alturas menores a 15m además de obtener menor precisión, tiene mayor riesgo de que el instrumento sea sabotado desde tierra.

Es necesario tomar en cuenta que el UAV trabaja a diferentes alturas, mediante los resultados que se obtuvieron en la experimentación se puede decir que es confiable utilizar el dron para la detección a un rango de altura de 20m hasta 50m ya que en ese rango de distancia desde el suelo los resultados serán mas confiables. Se descarta el de 15m por 2 desventajas ya que al estar a una altura mas baja se tiene menor precisión y a esa distancia es mas fácil que el dron sea sabotado por personal en tierra.

Como se evidenció en el estado del arte, existen múltiples trabajos que muestran el uso de redes neuronales convolucionales, para determinar aglomeraciones en el contexto del COVID-19. Sin embargo, sería importante analizar y comparar los rendimientos de estos modelos y su costo computacional, ya que son dos parámetros importantes para que los sistemas tengan éxito en aplicaciones de la vida real.

Existen sistemas más avanzados podrían necesitar mayores requerimientos lo que supone un encarecimiento del sistema. Siendo un punto de partida de una gran línea de investigación, este trabajo abre campo a muchas mejoras y trabajos de optimización del sistema. Pudiendo utilizar alguna otra metodología de empleo de redes neuronales para la detección de las personas, así como también el uso de otras técnicas para la determinación de las distancias en el escenario, en fin así se puede lograr que este sistema sea cada vez más robusto y que pueda ser empleado para la colaboración con la sociedad en estos tiempos difíciles de lucha contra la pandemia.

Referencias

- M. Aghaei, M. Bustreo, Y. Wang, G. Bailo, P. Morerio, and A. Del Bue. Single image human proxemics estimation for visual social distancing. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2784–2794, 2021. doi: 10.1109/WACV48630.2021.00283.
- A. H. Ahamad, N. Zaini, and M. F. A. Latip. Person detection for social distancing and safety violation alert based on segmented roi. In *2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pages 113–118, 2020. doi: 10.1109/ICCSCE50387.2020.9204934.
- D. Babu Sam, S. Surya, and R. Venkatesh Babu. Switching Convolutional Neural Network for Crowd Counting, 2017.
- A. Barnawi, P. Chhikara, R. Tekchandani, N. Kumar, and B. Alzahrani. Artificial intelligence-enabled internet of things-based system for covid-19 screening using aerial thermal imaging. *Future Generation Computer Systems*, 2021.
- K. Bhambani, T. Jain, and K. A. Sultanpure. Real-time face mask and social distancing violation detection system using yolo. In *2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC)*, pages 1–6, 2020. doi: 10.1109/B-HTC50970.2020.9297902.
- F. Bouhleb, H. Mliki, and M. Hammami. Crowd behavior analysis based on convolutional neural network: Social distancing control covid-19. In *VISIGRAPP (5: VISAPP)*, pages 273–280, 2021.
- C. Chalmers, P. Fergus, C. Aday Curbelo Montanez, S. Longmore, and S. Wich. Video analysis for the detection of animals using convolutional neural networks and consumer-grade drones. *Journal of Unmanned Vehicle Systems*, 9(2):112–127, 2021. doi: 10.1139/juvs-2020-0018. URL <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85107386549&doi=10.1139%2fjuvs-2020-0018&partnerID=40&md5=722af2d3946c643034016e16be038fe4>. cited By 1.
- V. Chamola, V. Hassija, V. Gupta, and M. Guizani. A comprehensive review of the covid-19 pandemic and the role of iot, drones, ai, blockchain, and 5g in managing its impact. *IEEE Access*, 8:90225–90265, 2020. doi: 10.1109/ACCESS.2020.2992341.
- A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2008. doi: 10.1109/CVPR.2008.4587569.
- K. Chen, C. C. Loy, S. Gong, and T. Xiang. CHEN, LOY, GONG, XIANG: FEATURE MINING FOR LOCALISED CROWD COUNTING Feature Mining for Localised Crowd Counting. 2021. URL <http://www.eecs.qmul.ac>.

- C. Conte, G. de Alteriis, F. De Pandi, E. Caputo, R. S. Lo Moriello, G. Rufino, and D. Accardo. Performance analysis for human crowd monitoring to control covid-19 disease by drone surveillance. In *2021 IEEE 8th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pages 31–36, 2021. doi: 10.1109/MetroAeroSpace51421.2021.9511671.
- H. Cruz, R. P. R. Ch, and M. Pinillos. Automatic counting of people in crowded scenes, with drones that were applied in internal defense operations on october 20, 2019 in ecuador. In *International Conference of Research Applied to Defense and Security*, pages 111–121. Springer, Singapore, 2020.
- S. Degadwala, D. Vyas, H. Dave, and A. Mahajan. Visual social distance alert system using computer vision amp; deep learning. In *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 1512–1516, 2020. doi: 10.1109/ICECA49313.2020.9297510.
- K. Doull, C. Chalmers, P. Fergus, S. Longmore, A. Piel, and S. Wich. An evaluation of the factors affecting ‘poacher’ detection with drones and the efficacy of machine-learning for detection. *Sensors*, 21(12), 2021. doi: 10.3390/s21124074. URL <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85107676942&doi=10.3390%2fs21124074&partnerID=40&md5=94d238a67130e2a5b22dcab2129d6030>. cited By 0.
- B. B. Gawde. A fast, automatic risk detector for covid-19. In *2020 IEEE Pune Section International Conference (PuneCon)*, pages 146–151, 2020. doi: 10.1109/PuneCon50868.2020.9362389.
- S. Gupta, R. Kapil, G. Kanahasabai, S. S. Joshi, and A. S. Joshi. Sd-measure: A social distancing detector. In *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 306–311, 2020. doi: 10.1109/CICN49253.2020.9242628.
- P. ICHI. Una breve historia sobre la detección de objetos más rápida de R-CNN, 2021. URL <https://ichi.pro/es/una-breve-historia-sobre-la-deteccion-de-objetos-mas-rapida-de-r-cnn-91852028121230>.
- D. Kang, Z. Ma, and A. B. Chan. Beyond counting: Comparisons of density maps for crowd analysis tasks—counting, detection, and tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(5):1408–1422, 2019. doi: 10.1109/TCSVT.2018.2837153.
- A. Kumar, K. Sharma, H. Singh, S. G. Naugriya, S. S. Gill, and R. Buyya. A drone-based networked system and methods for combating coronavirus disease (covid-19) pandemic. *Future Generation Computer Systems*, 115:1–19, 2021. ISSN 0167-739X. doi: <https://doi.org/10.1016/j.future.2020.08.046>. URL <https://www.sciencedirect.com/science/article/pii/S0167739X20317064>.
- D. Lian, J. Li, J. Zheng, W. Luo, and S. Gao. Density Map Regression Guided Detection Network for RGB-D Crowd Counting and Localization, 2019.
- J. Liu, C. Gao, D. Meng, and A. G. Hauptmann. DecideNet: Counting Varying Density Crowds Through Attention Guided Detection and Density Estimation, 2018.
- S. Madane and D. Chitre. Social distancing detection and analysis through computer vision. In *2021 6th International Conference for Convergence in Technology (I2CT)*, pages 1–10, 2021. doi: 10.1109/I2CT51068.2021.9418195.
- N. H. Motlagh, M. Baga, and T. Taleb. Uav-based iot platform: A crowd surveillance use case. *IEEE Communications Magazine*, 55(2):128–134, 2017. doi: 10.1109/MCOM.2017.1600587CM.
- F. A. Naqiyuddin, W. Mansor, N. M. Sallehuddin, M. N. S. Mohd Johari, M. A. S. Shazlan, and A. N. Bakar. Wearable social distancing detection system. In *2020 IEEE International RF and Microwave Conference (RFM)*, pages 1–4, 2020. doi: 10.1109/RFM50841.2020.9344786.
- F. Outay, H. A. Mengash, and M. Adnan. Applications of unmanned aerial vehicle (uav) in road safety, traffic and highway infrastructure management: Recent advances and challenges. *Transportation Research Part A: Policy and Practice*, 141:116–129, 2020. ISSN 0965-8564. doi: <https://doi.org/10.1016/j.tra.2020.09.018>. URL <https://www.sciencedirect.com/science/article/pii/S096585642030728X>.

- M. Paredes, M. Calderón, J. Riquelme, and M. Darwin. Multipurpose unmanned system: an efficient solution to increase the capabilities of the uavs. In *2020 IEEE ANDESCON*, pages 1–6, 2020. doi: 10.1109/ANDESCON50619.2020.9272196.
- M. Rashied Hussein, A. B. Shams, E. Hoque Apu, K. Abdullah Al Mamun, and M. Shahriar Rahman. Digital Surveillance Systems for Tracing COVID-19: Privacy and Security Challenges with Recommendations. *arXiv e-prints*, art. arXiv:2007.13182, July 2020.
- Z. Shao, G. Cheng, J. Ma, Z. Wang, J. Wang, and D. Li. Real-time and accurate uav pedestrian detection for social distancing monitoring in covid-19 pandemic. *IEEE Transactions on Multimedia*, pages 1–1, 2021. doi: 10.1109/TMM.2021.3075566.
- Y. Siriwardhana, C. De Alwis, G. Gür, M. Ylianttila, and M. Liyanage. The fight against the covid-19 pandemic with 5g technologies. *IEEE Engineering Management Review*, 48(3):72–84, 2020.
- P. Somaldo, F. A. Ferdiansyah, G. Jati, and W. Jatmiko. Developing smart covid-19 social distancing surveillance drone using yolo implemented in robot operating system simulation environment. In *2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC)*, pages 1–6, 2020. doi: 10.1109/R10-HTC49770.2020.9357040.
- S. Srinivasan, R. Rujula Singh, R. R. Biradar, and S. Revathi. Covid-19 monitoring system using social distancing and face mask detection on surveillance video datasets. In *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pages 449–455, March 2021. doi: 10.1109/ESCI50559.2021.9396783.
- M. Tzelepi and A. Tefas. Human crowd detection for drone flight safety using convolutional neural networks. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 743–747, 2017. doi: 10.23919/EUSIPCO.2017.8081306.
- C. Ye, H. Ling, F. Zou, and Z. Lu. Social networks based fingerprint code: Snbfc and its pirates tracing algorithm to majority attack. In *2010 International Conference on Multimedia Information Networking and Security*, pages 345–350, 2010. doi: 10.1109/MINES.2010.79.