BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN
INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

”DESARROLLO DE UN ALGORITMO BASADO EN TÉCNICAS DE APRENDIZAJE DE MÁQUINA PARA EL RECONOCIMIENTO DE ROSTROS Y SILUETAS EN TAREAS DE VIDEOVIGILANCIA”

Presenta:

JUAN CARLOS HERRERA PRO

Tesis presentada para obtener el grado de:
LICENCIATURA EN INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Director de Tesis:
Dr. IVAN OLMOS PINEDA

Agosto, 2021

Dedicatoria

A mi familia

Agradecimientos

Agradezco la paciencia de mis padres y su infinito apoyo en mi formación académica. Así como a mi asesor, el profesor Iván por su tiempo y dedicación.

Índice general

Índice de figuras

Capítulo 1

Introducción

1.1. Antecedentes

Históricamente la humanidad ha tenido la necesidad de avanzar, crear, solucionar, innovar, etc. Hoy en día, aún con todos los avances que existen no es suficiente para nosotros, los estudios realizados en materia de inteligencia artificial (IA) son sumamente notables. En nuestra vida cotidiana nos topamos con referencias a la IA y en muchos casos la utilizamos sin siquiera darnos cuenta, como podría ser al navegar por internet, al utilizar un filtro fotográfico en alguna red social o al buscar el mejor precio para un vuelo. También, es de hacerse notar que los avances en el hardware que poseemos para realizar dichas tareas cada día es más significativo. Esto abre el paso para aplicaciones más complejas en el ámbito de la IA. Pues si bien muchas de las técnicas que se consideran "nuevas" no lo son, ya que tienen un fundamento teórico de hace ya bastante tiempo, debido a limitaciones de hardware sus aplicaciones eran difíciles de realizar. En nuestros días la IA se ha diversificado en muchos campos, hasta el punto donde se ha convertido en una herramienta indispensable en el día a día de las personas. Una de las aplicaciones donde ha tenido principal auge, es en el reconocimiento de objetos, en especial, el de reconocimiento de figuras humanas, así como sus rostros. Actualmente se tienen sistemas muy complejos para la detección de estos. Gracias a los avances mencionados, el reconocimiento de personas juega un papel muy importante en muchos ámbitos, incluido el orientado a la seguridad donde por medio de cámaras de videovigilancia se podrían identificar siluetas de personas no deseadas o sus rostros. En la actualidad las tareas de videovigilancia comúnmente están realizadas por una simple cámara de video que graba lo ocurrido en alguna área especifica de una casa, una calle, etc. Estos son más bien medios pasivos pues no hacen más que registrar lo sucedido, y aunque en algunos casos podemos encontrar sistemas de cámaras de seguridad que implementan sensores, para detectar movimiento y así lanzar una alerta, estos presentan un inconveniente y es el de lanzar falsas alertas cuando, por ejemplo una mascota es detectada por el sensor. Debido a esto, la implementación de sistemas de seguridad con inteligencia artificial es una solución que podría presentar grandes beneficios pues nos darían resultados mucho más precisos y evitan las falsas alarmas y hasta podrían ayudar a la identificación de personas que están por cometer algún delito. Sobre todo, en países con alta incidencia de delincuencia, como lo son México, donde según la Fiscalía General del estado de Puebla, entre enero del 2020 a febrero del 2021, se registraron 2,537 denuncias por robo a casa-habitación, de las cuales, 2,077 fue sin violencia. [13].

Según cifras del INEGI en la Encuesta Nacional de Victimización y Percepción sobre Seguridad Pública 2020, el tipo de delitos que más se cometen, presentan un escenario donde el uso de nuevas tecnologías de videovigilancia, como podría ser la aplicación de IA, presentan grandes ventanas de oportunidad para dar un nuevo enfoque en el

combate a la delincuencia, ver figura 1.1.1.

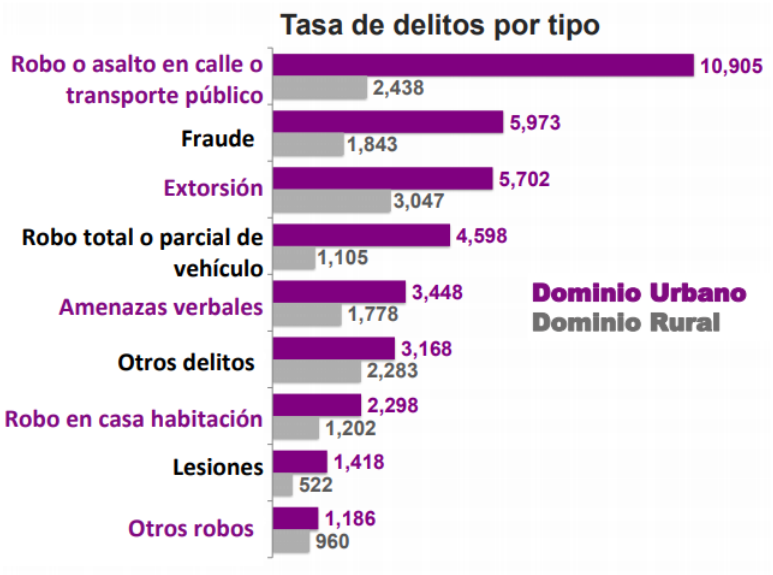


Figura 1.1.1: Tasa de delitos por tipo

Aunque esto presenta varios retos, pues si bien existen sistemas de reconocimiento de siluetas y rostros, estos constan de complejas cámaras y sistemas de servidores para poder funcionar, sin mencionar el alto coste de implementación de estos, esto hace que su implementación sea inviable para muchos escenarios. Por lo tanto, es de particular interés la investigación y desarrollo de algoritmos que presenten sistemas fiables basados en técnicas aprendizaje de máquina, capaces de detectar rostros y siluetas de personas, pues como se ha dicho la mayoría de los sistemas actuales presentan grandes inconvenientes que pueden resultar en falsas alarmas o en general un funcionamiento no deseado, además esta solución debe mantener un bajo costo de implementación, ya que como se ha mencionado, muchos de los sistemas de videovigilancia más avanzados son sumamente costosos lo que hace inviable su uso para ambientes domésticos.

1.2. Planteamiento del problema

Se hace notorio el índice de violencia que vive el país siendo este un grave problema para la vida diaria de los habitantes y también para el crecimiento de la nación. Este hecho, además, presenta cifras desalentadoras para años venideros, pues si bien a inicios del 2020, precisando el mes de enero, se registró un decrecimiento en el indicé de homicidios según las autoridades mexicanas de únicamente el 2.3 % y pese a ser menos que el año anterior, 2,819 víctimas sigue siendo una cifra alarmante. Para ejemplificar mejor lo dicho, se puede observar la figura 1.2.1.

Pero los homicidios no son el único delito que alerta a la población, ya que la privación de la libertad también es un tema grave en México, según datos del Secretariado Ejecutivo del Sistema Nacional de Seguridad Pública (SESNSP), del 2015 al 2019 se registró un aumento de 27 % de casos de secuestro registrados en México y tan sólo de enero a noviembre del año 2019 hubo 1,505 víctimas de privación ilegal de la libertad. Todas estas cifras son casos alarmantes pero también lo son las cifras que encontramos al tratar con el de caso a robo a negocio que, para tener noción del crecimiento de este, en todo el mes de mayo del año 2000 se registraron 4,442 carpetas de investigación por este delito, para mayo del 2019 se alcanzaron cifras de más del doble al contabilizarse 9,854

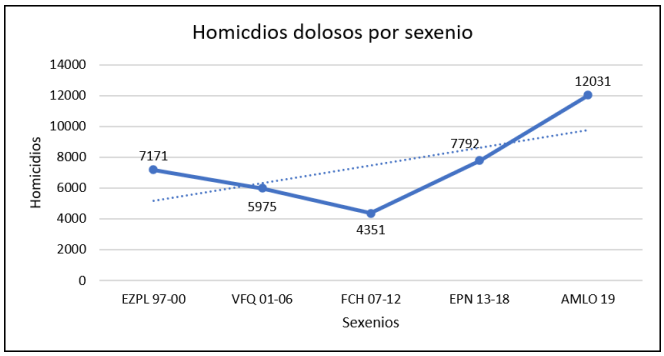


Figura 1.2.1: Homicidios dolosos por sexenio

carpetas, es decir 121 % más que hace casi 20 años y para resaltar aún más la alza de violencia, durante el primer semestre del año 2019 ya suman 47,441 carpetas de investigación por robo a negocio, 10.8 % más que en el mismo periodo del 2018, cuando se registraron 42,789. El robo a negocios presenta no solo un grave problema para las personas al poner en riesgo sus vidas, pues también presenta un duro golpe a la economía de cada individuo afectado y en general a la del país. Otro punto importante es el robo o asalto en la calle, así como en el transporte público, que según cifras del ENVIPE 2020, durante 2019 se cometieron 8.2 millones de robos o asaltos en calle o transporte público, lo cual representa una tasa de 9,091 robos cada 100 ,000 habitantes, ver figura 1.2.2. El 66.5 % de los casos, lo robado fue dinero, tarjetas de crédito o cheques, mientras que en 55.5 % de los casos fueron teléfonos celulares.

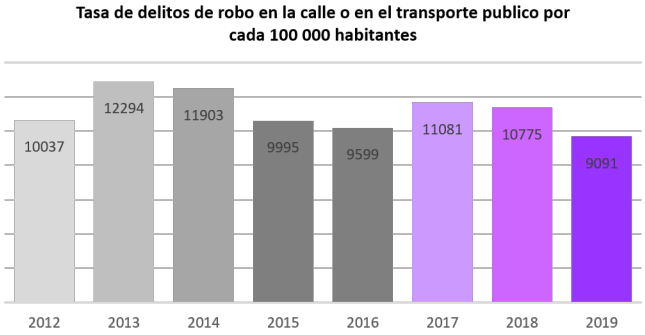


Figura 1.2.2: Tasa de delitos de robo en la calle o en el trasporte público cada 100,000 habitantes

Este tipo de delitos sin duda presenta uno de los problemas más graves que enfrenta el país, pues provoca que la población viva con un miedo constante. Aunque en este trabajo nos centraremos en otro tipo de delitos que también tienen una importante incidencia, este es el robo a casa habitación. Para darnos una idea del alto índice que tiene este delito durante el 2019 se cometieron 3500 de estos delitos por cada 100000 habitantes.

Este tipo de delitos también presenta un índice más alto de perdida por parte de los afectados, pues es obvio que algunos de los objetos de mayor valor monetario, se encuentren en la vivienda. Pero no solo es la perdida de los objetos robados el costo de un delito de este tipo, también lo son la inversión en medidas preventivas, pues según datos del ENVIPE 2020, el costo de medidas preventivas para evitar este tipo de robos es del 33.6 % del total de perdidas. Incluso podríamos llegar a considerar los gastos médicos consecuentes. Para tener noción de lo anteriormente mencionado, referirse las figuras 1.2.3 y 1.2.4.

Año	Costo total del delito		Costo de medidas preventivas (Miles de millones de pesos)
	Como porcentaje del PIB	Miles de millones de pesos	
2012	1.34	283	72.6
2013	1.27	269.4	80.9
2014	1.27	275.4	77.3
2015	1.25	281.7	92.7
2016	1.1	263.7	94.3
2017	1.65	322.9	96
2018	1.54	294.4	104.9
2019	1.53	282.1	94.8

Figura 1.2.3: Costo total a consecuencia del delito en hogares por año

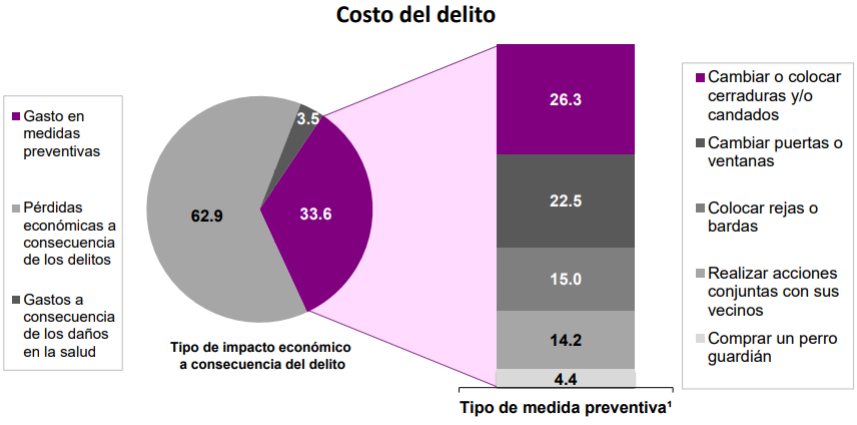


Figura 1.2.4: Distribución de los gastos a consecuencia del robo a casa habitación

1.3. Objetivos Generales y Específicos del Proyecto.

1.3.1. Objetivo general

Desarrollar un algoritmo para el reconocimiento de rostros y siluetas orientado a la videovigilancia, mediante el uso de algoritmos de reconocimiento de patrones implementando aprendizaje de máquina, el cual pueda ser implementado en sistemas de seguridad.

1.3.2. Objetivos específicos

- Investigar el estado del arte y los trabajos relacionados con este proyecto para realizar una tabla comparativa que mida el rendimiento de los algoritmos en términos de precisión, tiempo de procesamiento, recursos requeridos
- Proponer una metodología basada en visión artificial según lo investigado, capaz de reconocer rostros y siluetas operando en entornos de videovigilancia
- Implementar el algoritmo propuesto utilizando algún lenguaje de programación de propósito general
- Medir la precisión el algoritmo propuesto y compararlo con los ya investigados en ambientes de poca iluminación o con grabaciones de poca resolución

1.4. Hipótesis

El uso de inteligencia artificial como una solución viable a ciertos problemas es sin lugar a duda uno de los procesos de mayor coste computacional, muestra de ello es que la implementación de muchas de sus aportaciones no fue posible hasta la llegada de hardware con un poder de procesamiento considerable (véanse las redes neuronales artificiales, explicadas en la sección 2.4.2). Es por ello que los resultados esperados al implementar determinados algoritmos basados en el aprendizaje profundo tengan un bajo rendimiento. La tarea fundamental de esta labor de investigación, es la de orientar estas tecnologías a su uso con un bajo coste de implementación se espera, que al ir variando ciertos parametros como podrían ser la resolución de los vídeos, así como sus cuadros por segundo, el tiempo de procesamiento mejore. Pues si bien tecnologías como el aprendizaje profundo han presentado soluciones optimas en los campos de procesamiento de imágenes digitales, estos también se ven plagados por el recelo de la comunidad científica por su alto coste de implementación, ya que estos requieren de un hardware especializado (unidades gráficas de procesamiento) para tener resultados aceptables, si bien este tipo de hardware se ha abaratado con el paso del tiempo, las aplicaciones de aprendizaje profundo son cada vez más grandes y complejas. Es por esto que se esperan obtener mejores resultados al implementar algoritmos basados en aprendizaje profundo, pero a su vez, estos incrementen considerablemente el tiempo de procesamiento requerido comparándose con algoritmos basados en aprendizaje de maquina convencional.

1.5. Alcances y limitaciones

Con lo visto en el planteamiento del problema se esclarece uno de los principales alcances que tiene esta investigación, siendo un punto fundamental restringir el uso de los algoritmos de reconocimiento de rostros y siluetas en

áreas interiores y de uso doméstico. Esto porque, por ejemplo, dentro de una casa habitación se tiene un ambiente más "controlado" además de que las cámaras de videovigilancia que suelen emplearse en este tipo de escenarios son más favorables para el reconocimiento de rostros y siluetas por tratar con distancias no muy grandes, ya sea una sala de estar, una habitación e incluso un pasillo. Lo anterior mencionado, se definiría como el principal alcance, tener un sistema que de manera óptima reconozca rostros y siluetas en áreas pequeñas. Por parte de las limitaciones, estas estarían más bien ligadas al hardware utilizado, el cual no debe presentar una inversión considerable. Es por esto mismo que el uso de determinados algoritmos que puedan ser más eficientes o presentar soluciones interesantes en la tarea a tratar sean descartados por depender de un hardware más potente y por ende costoso.

Capítulo 2

Marco Teórico

Este capítulo tiene la intención de describir terminología, así como ejemplificar conceptos que serán de utilidad para una mejor compresión de temas puntuales en capítulos posteriores. Así mismo, dar planteamiento para algunas formulas que se desarrollan en esta sección.

2.1. Terminográfia

Cuadros por segundo
Es la frecuencia (tasa) con la cual un dispositivo es capaz de mostrar o capturar imágenes, conocidas como fotogramas a lo largo de un segundo. A considerarse que hablaremos de forma indistinta entre cuadros por segundo y FPS (del inglés Frames per Second).

Fotogramas
Es una imagen fotográfica, en términos estrictos esta se obtiene a través de la colocación de objetos por encima de una superficie fotosensible como por ejemplo una película o papel fotográfico y posteriormente a la exposición a la luz directa. Aunque para nosotros el termino "fotograma" hará referencia a una de las tantas imágenes de las que se compone un video es por ello que en este documento se hablará de forma indistinta entre fotograma, frame o cuadro.

Pixeles
La palabra píxel surgió hace años como acrónimo del concepto en inglés (Picture Element), que en castellano significa elemento de imagen, y consiste básicamente en la más pequeña unidad homogénea en color que compone la imagen digital.

Modelo de color RGB
El espacio de color RGB está formado por los colores luz primarios: Rojo, Verde y Azul. RGB son las siglas en inglés de Red, Green y Blue. Este sistema es el más adecuado para representar las imágenes que se muestran en monitores y que, finalmente pueden ser impresas papel fotográfico Las imágenes RGB utilizan tres colores para reproducir en pantalla hasta 16,7 millones de colores.

Profundidad del color
Cuando se habla de profundidad de color, el concepto se refiere puntualmente a la forma de codificación de cada uno de los pixeles que conforman una imagen, y que en definitiva brinda la posibilidad de lograr una completa gama

de tonalidades. Para ello, en las imágenes de mapa de bits cada uno de los pixeles es codificado individualmente por intermedio de un conjunto de bits de longitud determinada. Por lo general, en el terreno de la fotografía digital suelen emplearse tres bytes para definir un color, por lo que es posible representar 256 colores, los cuales logran sumar 16.777.216 opciones de color.

Resolución de imagen

La resolución de una imagen no es un concepto único, sino que depende del medio en que la imagen vaya a ser tratada o visualizada. El concepto más conocido es la resolución del archivo digital, la densidad de píxeles en la imagen definida por el número de píxeles distintos que tiene la imagen por unidad de longitud. Su unidad de medida son los PPI/PPP (pixeles per inch/píxeles por pulgada), esto se refiere a la resolución espacial . Cuanto mayor sea esta resolución, más contenedores de información (píxeles) tiene el fichero digital, más calidad tendrá la imagen y más peso (bytes) tendrá el fichero. Para fines prácticos se utilizará un formato de pixeles por pixeles, es decir n x m pixeles para referirse a lo largo y ancho de pixeles que contenga una imagen.

Resolución espacial

La resolución espacial se refiere a la finura de detalles visibles en una imagen: cuanto menor es el área terrestre representada por cada píxel en una imagen digital mayores son los detalles que pueden ser captados y mayor es la resolución espacial. Existen diferentes factores que influyen en la resolución espacial: además de los obviamente asociados al sensor, como ser el poder resolutivo del sistema óptico, debemos considerar las influencias atmosféricas, presencia de humo, neblina, bajos niveles de iluminación, etc.

Tiempo de procesamiento por video

Este valor esta dado por el tiempo en que un algoritmo tarda en procesar todos los cuadros de un video, es decir que, si un video está configurado a 20 cuadros por segundo y a su vez tiene una duración de 10 segundos, quiere decir que el algoritmo tendrá que procesar un total de 200 cuadros. Este tiempo guarda una relación de proporción con los FPS de salida de cada video.

Cuadros por segundo promedio por video

Este valor es dado por el tiempo de procesamiento por video dividido por el número total de cuadros que tenga un video. De tal manera que podremos obtener la siguiente formula.

$$FPS = \frac{Ct}{Tp} \tag{2.1}$$

Donde:

$$Tp = Tiempo\ de\ procesamiento\,(segundos)$$

$$Ct = cuadros\ totales$$

$$FPS = Cuadros\ por\ segundo\ promedio$$

2.2. Cámaras de video vigilancia

En el mundo de las cámaras de video, precisamente en el de las cámaras de seguridad podemos encontrar un amplio rango de productos dedicados a satisfacer dicha tarea, aunque la configuración usual para la mayoría de estos dispositivos constan de cámaras capaces de grabar en una resolución HD (1280 x 720 píxeles) a 30 cuadros por segundo. Y a su vez estas cámaras pueden estar diseñadas para trabajar en interiores o exteriores. De esto último depende el tipo de lente que se utiliza, para el primer caso, es usual que se coloquen cámaras con un gran ángulo de visión que permitan tener una amplia visualización de una parte específica de algún cuarto, aunque teniendo una distancia efectiva reducida, por el contrario, para exteriores, lo común es utilizar cámaras con un ángulo de visión más corto, pero con una distancia efectiva mucho mayor, esto se ejemplifica el la figura 2.2.1.

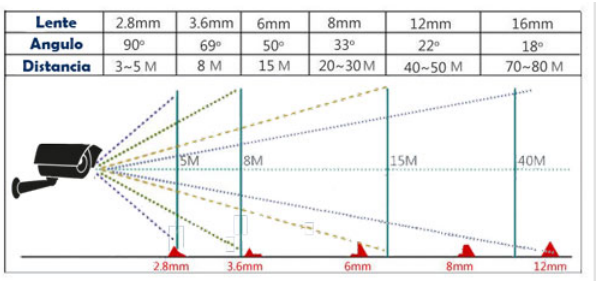


Figura 2.2.1: Relación de la distancia según el tipo de lente

Por lo general las cámaras de vigilancia constituyen un sistema de seguridad que consiste en realizar vigilancia a través de cámaras de video en diferentes lugares o ambientes. También es denominado circuito cerrado de televisión o CCTV (del inglés, closed circuit televisión) Esta tecnología de videovigilancia es especialmente útil para identificar intrusos o cualquier persona que realice alguna actividad indebida que ponga en riesgo la integridad de un lugar o individuo. El circuito está compuesto por una o varias cámaras de vigilancia que se interconectan a monitores de vídeo o televisores, los cuales reproducen las imágenes captadas por las cámaras o muchos de estos también son almacenados para mantener un registro que pueda ser útil en momentos determinados. Estas cámaras fijas permiten realizar una monitorización en remoto de personas, materiales y alarmas desde una sala de control central, desde la cual se configura el enfoque, la dirección, la inclinación o la vista panorámica. En la actualidad, estos sistemas de CCTV pueden incluir visión nocturna, térmica, o detección de movimiento

2.2.1. Cámaras de detección de movimiento

Un detector de movimiento con cámara o sensor de presencia con cámara es un aparato electrónico que dispone de unos sensores especializados que identifican el movimiento que ocurren delante de él. Por regla general, estos dispositivos activan un sistema de alarma cuando detectan dicho movimiento. Sólo graban cuando ese movimiento se efectúa, de forma que pueden contener más tiempo de vídeo útil. Existen numerosos tipos de detectores de movimiento con cámara y sin ella. En función del sistema que utilicen para detectar el movimiento encontramos:

- Detector de movimiento de rayos infrarrojos pasivo
- Detector de microondas
- Detector de movimiento dual de rayos infrarrojos y de microondas
- Detector que utiliza ultrasonidos

Prácticamente todo el mercado orientado a la videovigilancia trabaja la detección de movimiento con el uso de sensores, pues de momento son la opción para solucionar este problema más barata y simple de implementar.

2.2.2. Consideraciones FPS y ángulos de visión

Debido a las limitaciones de hardware que se podrían presentar para implementar cierto tipo de algoritmos para el reconocimiento de personas. Es por ello que se hace necesario hacer una estimación de la mínima cantidad de cuadros por segundo con la cual es posible realizar una detección, es decir sin perder por la falta de cuadros. Como la detección está orientada hacia el reconocimiento de seres humanos, el sistema deberá ser capaz de detectar humanos aún mientras estos estén en movimiento, pero al estar en movimiento es posible que alguno salga de cuadro durante algún intervalo de muestreo, es decir que, entre un fotograma a otro, una persona con la suficiente velocidad salga de la toma. Para evitar esto se ha realizado una estimación entre el número de veces que una persona será captada por una cámara si esta graba a determinados fotogramas, para esto se consideran cámaras con los tipos de lentes antes mencionados. A su vez se ha realizado una investigación de la velocidad con la que corre en promedio un ser humano.

La velocidad promedio de un ser humano es de 6.66 m/s (24 KM/h aproximadamente), tomando en cuenta lo anterior podemos obtener una fórmula que nos determinará la cantidad de cuadros aproximada en la que una persona aparecerá en cuadro

$$\frac{De}{v} \ * \ F \ = Cuadros \ registrados \tag{2.2}$$

Donde

$$De \ = \ Distancia \ efectiva \ de \ la \ camara \ (metros)$$

$$v \ = Velocidad \ promedio \ (metros \ / \ segundo)$$

$$F \ = \ Fotogramas \ por \ segundo$$

Por lo tanto, si quisiéramos saber en cuantos cuadros sería captada una persona que corre a una velocidad de 6.66 m /s captada por una cámara que tiene una distancia efectiva de 5 metros y que graba a 24 cuadros por segundo

$$\frac{5 \ m}{6{,}66 \ m/s} * 24 \ /s \ \rightarrow 0{,}75075 \ s \ * \ 24 \ /s \ \ = \ 18{,}01801 \tag{2.3}$$

Y debido a que los cuadros en video de referencian con numero enteros, podemos redondear a 18 cuadros en los que una persona moviéndose a esa velocidad es registrada. Para complementar lo anterior podemos referirnos a la figura 2.2.2.

Lente	Grados	Distancia efectiva	Cuadros
2.8 mm	90°	5	18.01801802
3.6 mm	69°	8	28.82882883
6 mm	50°	15	54.05405405
8 mm	33°	30	108.1081081
12 mm	22°	70	252.2522523
16 mm	18°	80	288.2882883

Figura 2.2.2: . Relación de una grabación a 24 cuadros por segundo y una persona moviéndose a 6.66 m/s

2.3. Procesamiento digital de imágenes

Cabe mencionar que las definiciones presentadas a continuación, provienen del libro *Deep learning for computer vision with Python* [1].

En este apartado se verán los conceptos básicos de análisis de imágenes y se introducirán los tipos de procesado más comunes. El procesamiento digital de Imágenes es el área de la ingeniería que se encarga de la extracción de mediciones, datos o información contenida en una imagen. Incluye aquellas técnicas cuyo principal objetivo es facilitar la búsqueda e interpretación de la información contenida en ellas. Un sistema de análisis de imágenes se distingue debido a que tiene como parámetro de entrada una imagen, y cuyo resultado es comúnmente una salida numérica, en lugar de otra imagen. Esta salida es la información referente al contenido de la imagen de entrada. Sin embargo, para llegar desde la imagen original al conjunto de parámetros e información extraída de la misma, es necesario pasar por distintas etapas de procesamiento y filtrado donde se analiza la imagen y se adecua para cierta aplicación específica. Esto implica que el resultado del procesamiento depende fuertemente del problema que se esté abordando. El procesamiento y análisis de imágenes se ha desarrollado en respuesta a tres de los más grandes problemas concernientes a imágenes:

- La digitalización y codificación de imágenes que facilite la transmisión, representación y almacenamiento de estas
- Mejora y restauración de una imagen para interpretar más fácilmente su contenido
- Descripción y segmentación de imágenes para aplicaciones de visión robótica o visión artificial

Todos aquellos algoritmos de procesamiento de imágenes destinados a resaltar, agudizar y/o contrastar determinados aspectos de la imagen, y también aquellos que ayudan a eliminar efectos no deseados sobre ellas, como toda clase de ruido (aditivo, sustractivo, multiplicativo, etc.), se denominan técnicas de mejora de la imagen El conjunto de métodos de procesamiento de imágenes está dividido en tres grandes grupos:

- Algoritmos en el dominio espacial. Se refiere a métodos que procesan una imagen píxel por píxel, o también tomando en cuenta un conjunto de píxeles vecinos.
- Algoritmos en el dominio de la frecuencia. Frecuentemente, estos métodos son aplicados sobre los coeficientes resultantes de la Transformada de Fourier de una imagen.
- Algoritmos de extracción de características. A diferencia de los dos grupos anteriores, los algoritmos de extracción de características están enfocados al análisis de imágenes para la extracción de atributos y regiones de interés, separación de objetos del fondo, detección de bordes o formas, entre otros.

2.3.1. Imagen digital

Una imagen se define como una función de dos dimensiones $f(x, y)$ donde x e y son las coordenadas de un plano que contiene todos los puntos de esta, y $f(x, y)$ es la amplitud en el punto (x,y) a la cual se le llama intensidad o nivel de gris de la imagen en ese punto. En el caso de que tanto las coordenadas x e y como los valores de intensidad de la función f sean discretos y finitos, se habla de una imagen digital.

Una imagen digital está compuesta de un número finito de elementos y cada uno tiene una localidad y un valor particular. A estos elementos se les llama puntos elementales de la imagen o píxeles, siendo este último el término comúnmente utilizado para denotar la unidad mínima de medida de una imagen digital.

En la figura 2.3.1 se muestra una representación de una imagen con 256 niveles de intensidad. En ella, cada uno de los píxeles está representado por un número entero que es interpretado como el nivel de intensidad luminosa en la escala de grises. Ampliando la imagen en una zona cualquiera, se pueden apreciar estos valores, que se muestran en forma de matriz en la misma figura, correspondiéndose cada elemento de la matriz P_{ij} con las coordenadas en el plano $x = i, y = j$.

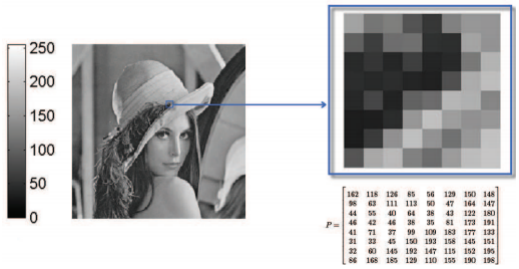


Figura 2.3.1: Representación de una imagen digital a una matriz numérica

2.3.2. Imágenes a color

El fundamento para describir una imagen digital en color es el mismo que el expuesto anteriormente, con la salvedad de que cada elemento o píxel es descrito y codificado de otra forma, según el espacio de color que se esté utilizando. Así, por ejemplo, para un espacio de color RGB (generalmente el más usado para representar imágenes), se representa cada píxel como un color creado a partir de ciertas cantidades de los colores rojo, verde y azul. Esta representación se puede interpretar como una matriz de tres niveles de intensidad, donde cada nivel corresponde a la intensidad de color de los componentes rojo, verde y azul, como se muestra en la figura 2.3.2

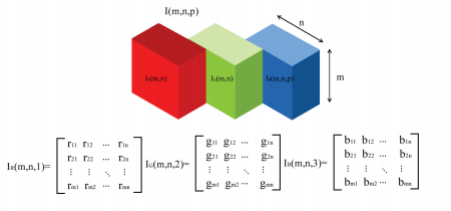


Figura 2.3.2: Representación del espacio de color RGB en matrices numéricas

En una imagen RGB, cada píxel está compuesto por un valor de intensidad correspondiente a cada componente primaria. El color resultante del píxel vendrá por tanto definido por la çantidad"de intensidad que tenga cada

componente. Así, el color blanco estará compuesto de la máxima intensidad de color para los tres componentes. Por el contrario, el color negro será el resultado de reducir al mínimo la intensidad de los componentes ver la figura 2.3.3

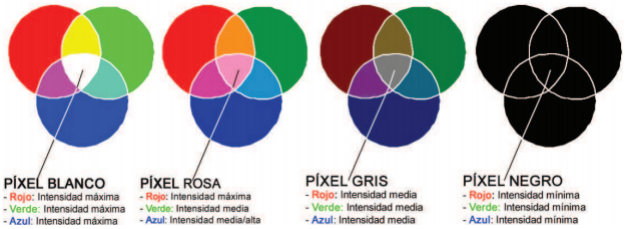


Figura 2.3.3: Forma en que el espacio de color RGB forma diferentes colores

2.3.3. Histogramas

Los histogramas son gráficos que indican la frecuencia de un hecho mediante una distribución de los datos, ejemplo en la figura 2.3.4. Los histogramas no se pueden elaborar con atributos, sino con variables medibles tales como peso, temperatura, tiempo, etc. Un histograma muestra la acumulación o tendencia, la variabilidad o dispersión y la forma de la distribución. Se trata de una gráfica adecuada para representar variables continuas, aunque también se puede usar para variables discretas. Es decir, mediante un histograma se puede mostrar gráficamente la distribución de una variable cuantitativa o numérica. Los datos se deben agrupar en intervalos de igual tamaño, llamados clases.

Figura 2.3.4: Ejemplo de un histograma

Se aplica a todos aquellos estudios en los que es necesario analizar la pauta de comportamiento de un determinado fenómeno en función de su frecuencia de aparición.

Por su naturaleza gráfica, el histograma puede ayudar a identificar e interpretar pautas que son difíciles de ver con una simple tabla de números y que son de poco valor si no aparecen suficientemente ordenados y clasificados.

Permite resumir grandes cantidades de datos y comunicar información clara y sencilla sobre situaciones complejas. Se aplica a todos aquellos estudios en los que es necesario analizar la pauta de comportamiento de un determinado fenómeno en función de su frecuencia de aparición y Permite:

1. Mostrar el patrón de variación.
2. Comunicar información visual acerca del comportamiento del proceso
3. Tomar decisiones acerca del punto en que se deben concentrar los esfuerzos para lograr la mejora

2.4. Inteligencia artificial

Si bien en la gran mayoría de las ciencias o diciplinas ocurre que se tiene una definición clara de conceptos, donde más o menos se delimitan sus alcances, así como su contenido desde el mismo comienzo de su estudio. Pues si bien se tienen una gran cantidad de ideas y conceptos relacionados con la inteligencia artificial, esta no dispone de una definición o un consenso claro entre la comunidad de científicos que se dedican al estudio de esta rama de la computación. Podríamos decir que este término dispone de un problema fundamental y parte del concepto de inteligencia mismo, pues este también es algo que suele ser mal definido resultando en una pobre comprensión del concepto.

Según lo avances realizados en esta rama, parece que se inclina únicamente a la solución de problemas que a nuestra consideración llamamos "complejos" como estimaciones numéricas, pero esto nos pone a pensar en el hecho de que existen cierto tipo de escenarios donde un "problema complejo" para una maquina no lo es para un ser humano en sí, como podría ser el reconocimiento de personas, que más que un problema para nosotros, resulta de una acción natural, dada por nuestras habilidades cognitivas de asociar una forma en especifica con un objeto, pues fácilmente podemos diferenciar entre una máscara de un rostro real. Este tipo de tareas son especialmente costosas para una máquina. Esto nos dice que realmente se busca dotar de un "conocer humano" a un ente que no es un humano. Según la referencia al libro *Inteligencia Artificial. Métodos, técnicas y aplicaciones* [23] presenta dos formas de entender este concepto.

La primera hace énfasis en la vía descendente y en el uso de conceptos del lenguaje natural (hechos y reglas) para representar el conocimiento necesario para resolver problemas de decisión que no necesitan un robot como componente imprescindible en su implementación.

La segunda hace énfasis en la vía ascendente y en el uso de conceptos de más bajo nivel semántico. Considera la inteligencia como una forma "superior" de adaptación al medio y se apoya en conductas y mecanismos implementables en un robot real que tiene que interactuar con un entorno real concreto Aún con todo esto no se hace debe oscurecer a los avances realizados en materia de inteligencia artificial, como podrían ser los realizados en la robótica, aprendizaje automático, visión artificial o el procesamiento del lenguaje natural, cada avance está acompañado de un o más campo de innovación e investigación, pues como cita el mismo libro
"Se han realizado avances importantes en las técnicas de modelado conceptual y formal, en la estructuración del conocimiento necesario para resolver una tarea en términos del "papel" que juegan los distintos elementos (roles) y del plan estratégico de descomposición del procedimiento de solución ("métodos"). También se ha avanzado en las técnicas de representación formal (lógica, reglas, marcos, objetos, agentes, redes causales, etc.) y en el tratamiento de la incertidumbre (redes bayesianas, sistemas borrosos) y en la solución de problemas para los que disponemos de más datos que conocimiento (redes de neuronas artificiales). Hay avances importantes en la búsqueda de inspiración en la biología (computación de membranas) y en la Física (computación cuántica)"

Por lo visto no podemos afirmar que la inteligencia artificial, persiga un objetivo estático y finito, si no que más bien busca, por medio de la adaptación de conceptos cognitivos, el apego a un tipo específico de inteligencia humano. Y hablando de estos conceptos cognitivos, podríamos apoyarnos en la consideración de Carreto Diaz [8], que menciona a los procesos cognitivos fundamentales, mencionar los siguientes:

- Memorización (Almacenamiento y recuperación de datos)
- Aprendizaje
- Solución de problemas

- Interferencia y deducción lógica
- Percepción y reconocimiento de formas
- Toma de decisiones
- Comprensión del lenguaje natural

Pues como se mencionó anteriormente, el autor también concuerda en que esta es solo una parte del comportamiento "inteligente" en los seres humanos.

Pues en este mismo sentido, se pronuncia Boden [4] afirmando: Sin embargo, una cosa es cierta: La inteligencia artificial no es el estudio de las computadoras. Las computadoras son máquinas metálicas de interés intrínseco para la ingeniería electrónica (...). Por inteligencia artificial, en consecuencia, entiendo el uso de programas de computadora y de técnicas de programación para proyectar luz sobre los principios de la inteligencia artificial en general y de la inteligencia humana en particular (...).
Para Sloman [32], la mejor forma de definir el concepto de inteligencia artificial es enumerando sus propósitos en términos generales. Así el distingue:

(...) tres finalidades principales en el dominio de la inteligencia artificial: análisis teórico de posibles explicaciones efectivas del comportamiento inteligente; explicación de habilidades humanas y construcción de artefactos inteligentes Para este autor, el comportamiento inteligente está íntimamente ligado con las habilidades para: Construir, interpretar, describir, modificar, comparar y utilizar estructuras complejas, incluidas las simbólicas, por lo que el estudio y avance en el campo de la inteligencia artificial y en su definición misma, digamos que sería directamente proporcional a el estudio y avance que se tenga en áreas como la psicología, la fisiología, la lingüística y la antropología.

2.4.1. Aprendizaje automático

Uno de los fines de las ciencias en general es entender y explicar cierto tipo de fenómenos, muchos de los cuales están representados por fórmulas matemáticas. La principal ventaja de contar funciones matemáticas es la probabilidad de predecir el comportamiento natural de un determinado fenómeno o sistema, de esta forma, si se trata de un sistema bastaría con seleccionar las entradas adecuadas para obtener una salida deseada, pero en el caso de mediante los cuales según las entradas podremos tener una salida concreta y en caso de los fenómenos, podremos al menos simular escenarios, algunos ejemplos serían la predicción de los valores en la bolsa, la meteorología o la clasificación de determinadas formas.

Aunque como menciona el autor R. D. Adrian [1], en numerosas ocasiones nos encontramos con escenarios diferentes.

"Muchas veces, sin embargo, no podremos dar detalles de ese proceso. El objetivo, en este caso, será estimar el modelo subyacente que genera los datos observados. En estos casos, las técnicas de aprendizaje automático nos permiten establecer estos modelos utilizando datos de ejemplo o experiencias pasadas."

Precisamente esto último es lo que trataría de aprendizaje automático, identificar ciertos patrones o regularidades en los datos y de esta manera dar aproximaciones precisas del problema a resolver.

2.4.2. Redes neuronales artificiales

Las redes neuronales artificiales (también llamadas RNA) son algoritmos pertenecientes a un campo de la Inteligencia artificial llamado, aprendizaje de máquina, especializados en aprender y reconocer patrones. Estas redes están inspiradas por la estructura y el funcionamiento del cerebro:

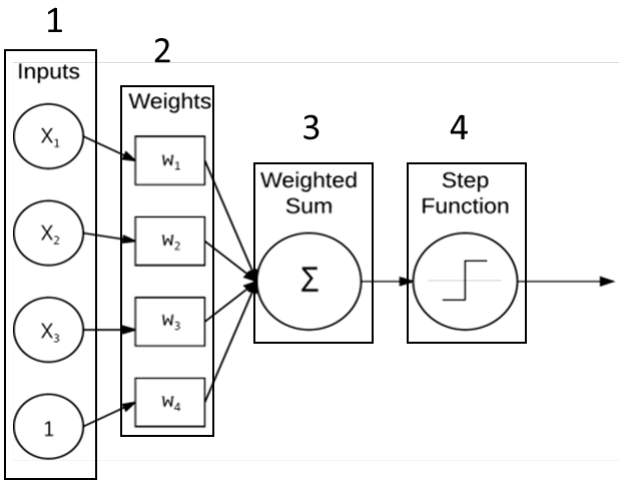


Figura 2.4.1: Representación de un perceptrón simple según el modelo de McCulloch y Pitts

Las redes neuronales actuales se basan en el modelo matemático de neurona propuesto por McCulloch y Pitts en 1943 [22] En dicho modelo cada neurona recibe un conjunto de $\{x_1, x_2, \dots x_D\}$ y devuelve una única salida, ver figura 2.4.1. Además, dentro de una RNA existen numerosas conexiones entre las distintas neuronas que la forman. Estas conexiones simulan las conexiones interneuronales del cerebro y, al igual que éstas, pueden establecerse con mayor o menor intensidad. En el caso de las RNA esta intensidad la determinan los pesos sinápticos (o simplemente pesos). De este modo, cada entrada $x_i de$ una neurona se encuentra afectada por un peso w_i Para obtener la salida y de la neurona es se debe calcular una suma ponderada de las entradas x_i con los pesos w_i, esto tiene el nombre de activación de la neurona:

$$a = \sum_{i=1}^{D} w_i x_i + w_0 \tag{2.4}$$

Donde w_0 es un umbral o sesgo que se utiliza para compensar la diferencia entre el valor medio de las entradas, sobre todo conjunto de entrenamiento y el correspondiente valor medio de las salidas deseadas. Posteriormente, con el valor y una vez obtenido de a, se aplica una función, llamada función de activación o de transferencia $g(a)$, es decir:

$$y = g\left(a\right) = g\left(\sum_{i=1}^{D} w_i x_i + w_0\right) = g\left(\sum_{i=0}^{D} w_i x_i\right) \tag{2.5}$$

Ahora la función de activación empleada en este modelo básico inspirado en el de McCulloch-Pitts es la función escalón definida por la ecuación:

$$g\,(a) = \{0\ cuando\ a < 0\ 1\ cuando\ \ a > 0\ \} \tag{2.6}$$

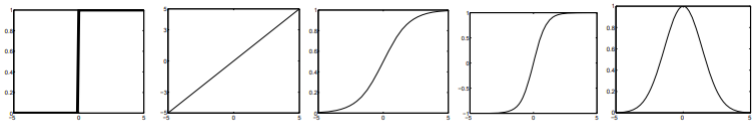
Actualmente, son utilizados también otro tipo de funciones de activación, las más comunes se mencionan a continuación

Lineal: $g\,(a) = a$

Sigmoide : $g\,(a) = \frac{1}{1+\,e^{-a}}$

Tangente hiperbólica (tanh): $g\,(a) = tanh = \frac{e^{a}-\,e^{-a}}{e^{a}+\,e^{-a}}$

Gaussiana: $g\,(a) = exp\left(-\frac{(a-\,\mu)^2}{2\sigma^2}\right)$



(a) escalón (b) lineal (c) sigmoidea (d) tanh (e) gaussiana

Figura 2.4.2: Gráficas de las diferentes funciones de activación

En la figura 2.4.2 se muestran las formas que tienen cada una de las funciones anteriormente descritas. Esto nos daría como resultado nuestro modelo básico de red neuronal completa con los elementos que vimos anteriormente, los cuales constan de las entradas x_i, los pesos w_i, la suma ponderada y la función que de activación. Al número de neuronas que componen una red neuronal, las conexiones que tienen unas con otras, se le conoce como *topología de la red*. Un ejemplo de esto se puede ver en la figura 2.4.3.

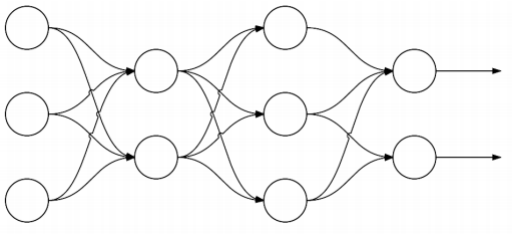


Figura 2.4.3: Topología de una red neuronal artificial

Aquí podemos ver un tipo de topología donde todas las neuronas están conectadas entre sí (al menos las que le proceden), en este caso contamos con 3 capas en nuestra red neuronal, pues la primera capa, que es la capa de entrada, la contaremos como la capa 0. (Figura 2.4.4).

Luego tenemos dos capas más, conocidas como capas ocultas, que serían las capas 1 y 2. (Figura 2.4.5).

Por último, nuestra última capa, se conoce como, capa de salida y corresponde a la capa 3. (Figura 2.4.6).

2.4.3. Entrenamiento de una red neuronal

Como se ha visto las RNA reciben datos de entrada, mismo que pasan por distintos procesos para producir una salida. La justificación teoría para estás aplicaciones es que, si se tiene una red neuronal con suficientes neuronas,

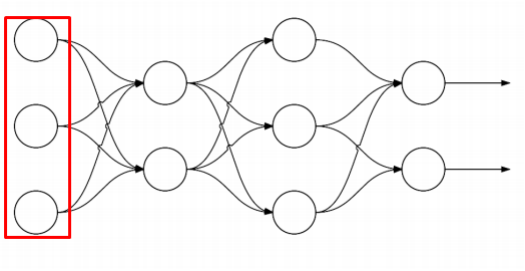


Figura 2.4.4: Capa de entrada de una red neuronal

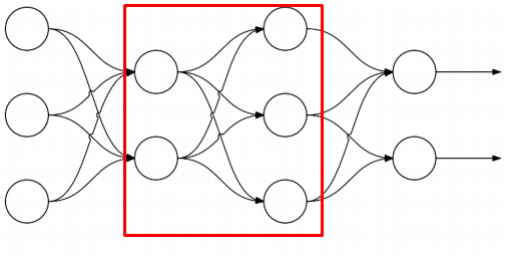


Figura 2.4.5: Capas ocultas de una red neuronal

esta RNA podrá ajustar cualquier función continúa alcanzando una precisión considerable únicamente con escoger los parámetros adecuados. Estos parámetros ajustables usualmente se refieren a los pesos sinápticos entonces podríamos decir que estos pesos son la forma que tiene la red para almacenar su conocimiento.

Este conocimiento se almacena en la RNA a través de un proceso de aprendizaje o entrenamiento, que no es más que la modificación de los parámetros de la RNA mediante un procedimiento establecido. Al igual que los seres humanos las RNA contemplan el uso de ejemplos que representan el problema. A este conjunto de ejemplos, se le conoce como “conjunto de entrenamiento”.

En este proceso de aprendizaje no se busca que una RNA memorice las relaciones que existen entre la entrada y la salida de una red, sino más bien, que modele el proceso que se genera con esos datos. Para ello se necesita que los conjuntos de entrenamiento sean representativos para la relación que se desea aprender. Por ello, una vez entrenada una red, no solo será capaz de manejar los datos con los que fue entrenada, también con datos distintos a los primeros. Esto se le conoce como “generalización” de la red.

2.4.4. Tipos de aprendizaje en una red neuronal

La forma que tienen de aprender las redes neuronales se ha clasificado en 3 tipos o mejor dicho, los algoritmos de aprendizaje de maquina suelen agruparse en 3 campos, según sea su tipo de aprendizaje:

- Aprendizaje supervisado: Nuestros datos de entrenamiento tienen asociadas etiquetas, por lo que podemos saber que datos pertenecen a que cosa. Se crea un modelo a través de un proceso de entrenamiento donde se hacen predicciones sobre los datos de entrada y luego se corrige cuando las predicciones son incorrectas. Este proceso de capacitación continua hasta que el modelo alcanza algún criterio de detención deseado, como una baja tasa de error o un número máximo de iteraciones de capacitación (Figura 2.4.7).
- Aprendizaje no supervisado: A diferencia del aprendizaje supervisado, el aprendizaje no supervisado (a veces llamado aprendizaje autodidacta) no tiene etiquetas asociadas con los datos de entrada y, por lo tanto, no

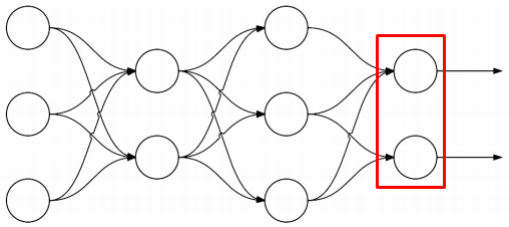


Figura 2.4.6: Capa de salida de una red neuronal

Etiqueta	R_μ	G_μ	B_μ	R_σ	G_σ	B_σ
Gato	57.61	41.36	123.44	158.33	149.86	93.33
Gato	120.23	121	181.43	154.58	69.13	116.91
Gato	124.15	193.35	65.77	23.63	193.74	162.7
Perro	100.28	163.82	104.81	19.62	117.07	21.11
Perro	117.43	22.31	149.49	197.41	18.99	187.78
Perro	149.73	87.17	187.97	50.27	87.15	36.65

Figura 2.4.7: Ejemplo de aprendizaje supervisado

podemos corregir nuestro modelo si hace una predicción incorrecta (Figura 2.4.8).

R_μ	G_μ	B_μ	R_σ	G_σ	B_σ
57.61	41.36	123.44	158.33	149.86	93.33
120.23	121	181.43	154.58	69.13	116.91
124.15	193.35	65.77	23.63	193.74	162.7
100.28	163.82	104.81	19.62	117.07	21.11
117.43	22.31	149.49	197.41	18.99	187.78
149.73	87.17	187.97	50.27	87.15	36.65

Figura 2.4.8: Ejemplo de aprendizaje no supervisado

- Aprendizaje semi-supervisado: Este tipo de aprendizaje, es un poco una combinación de ambas, donde solo algunos de las etiquetas de los datos (Figura 2.4.9)

2.4.5. Aprendizaje profundo

El término, Aprendizaje profundo está asociado con las redes neuronales artificiales, este término hace referencia al número de “capas” de las que se compone una red neuronal, es decir, su topología. Es necesario recalcar que no existe una definición ciento por ciento precisa para este término, pues es más bien la forma de nombrar a un concepto abierto a cambios y/o interpretaciones dentro del campo de la inteligencia artificial que varía dependiendo el autor.

Pues como menciona Jeff Dean en 2016 durante la conferencia, *Deep Learning for Building Intelligent Computer Systems :*

“Cuando escuches hablar del término aprendizaje profundo, solamente piensa en una gran red neuronal. Profundo se refiere usualmente al número de capas, pero, este terminó popular ha sido adoptado por la prensa”.

Como se mencionó anteriormente, que una red neuronal se considere “profunda” depende su topología, pero es preciso mencionar que tipo de topología debe tener una red para considerarse profunda. Esta es una pregunta que se hace recurrentemente en el estudio de las RNA pues como ya se dijo, este término está abierto a más de una interpretación. Lo que se plantea es converger en una definición. Para esto vamos a apoyarnos en la definición de. Adrian Rosebrock donde precisamente en su libro *Deep Learning for Computer Vision with Python* [1] menciona que una red neural es profunda si:

Etiqueta	R_μ	G_μ	B_μ	R_σ	G_σ	B_σ
Gato	57.61	41.36	123.44	158.33	149.86	93.33
?	120.23	121	181.43	154.58	69.13	116.91
?	124.15	193.35	65.77	23.63	193.74	162.7
Perro	100.28	163.82	104.81	19.62	117.07	21.11
?	117.43	22.31	149.49	197.41	18.99	187.78
Perro	149.73	87.17	187.97	50.27	87.15	36.65

Figura 2.4.9: Ejemplo de aprendizaje Semi-supervisado

1. Se tiene más de una capa en la red neuronal
2. Se está trabajando con un tipo de arquitectura especial (LSTM , redes neuronales convolucionales, etc.)

Con esto dicho, vamos a enfocarnos en una arquitectura especial de red neuronal, la cual está orientado a trabaja con imágenes.

2.4.6. Redes neuronales convolucionales

Son redes que tienen como principal propósito el procesamiento de imágenes

La RNC es un tipo de Red Neuronal Artificial con aprendizaje supervisado que procesa sus capas imitando al córtex visual del ojo humano para identificar distintas características en las entradas que en definitiva hacen que pueda identificar objetos y "ver". Para ello, la RNC contiene varias capas ocultas especializadas y con una jerarquía: esto quiere decir que las primeras capas pueden detectar líneas, curvas y se van especializando hasta llegar a capas más profundas que reconocen formas complejas como un rostro o la silueta.

Estas se caracterizan por poder aprender relaciones entrada-salida, donde la entrada es una imagen. Estas están basadas en operaciones de convolución y tiene como principal función:

- Detección/categorización de objetos
- Clasificación de escenas
- Clasificación de imágenes en general

Además, es necesario explicar las distintas capas que conforman una red neuronal convolucional

- Convolución (Conv)
- Activación (ReLu)
- Pooling

Convolución

Para ser más precisos se trata de una convolución de matrices y matemáticamente se describe como:

Dada una matriz $A_{m\ x\ n}$ y una matriz $C_{(2N+1)(2N+1)}$ con $2N+1 < m, n$ se define la convolución de matrices A y C como una matriz $D = A * C$ definida a partir de la expresión:

$$d_{ij} = \frac{1}{c} \sum_{r=1}^{2N+1} \sum_{s=1}^{2N+1} a_i - N + r - 1,\ i - N + r - 1\ Cr, S, \tag{2.7}$$

Donde $C = \sum_{i,j=1}^{2N+1} C_{ij}$ (si $C = 0$ se toma $c = 1$). Obsérvese que d_{ij} sólo está definido para $i = N+1, \ldots, m-N-1$ y $j = N+1, \ldots, n-N-1$

La matriz C se denomina núcleo o kernel de convolución.

Las figuras 2.4.10 y 2.4.11 muestran el proceso de convolución de una matriz dada por otra de orden 3 × 3. La Figura x recoge el resultado de la convolución correspondiente a las entradas (2, 2) y (3, 2). La Figura x proporciona el resultado final de la convolución en donde a las entradas de la primera y última fila y primera y última columna se les ha asignado el valor original.

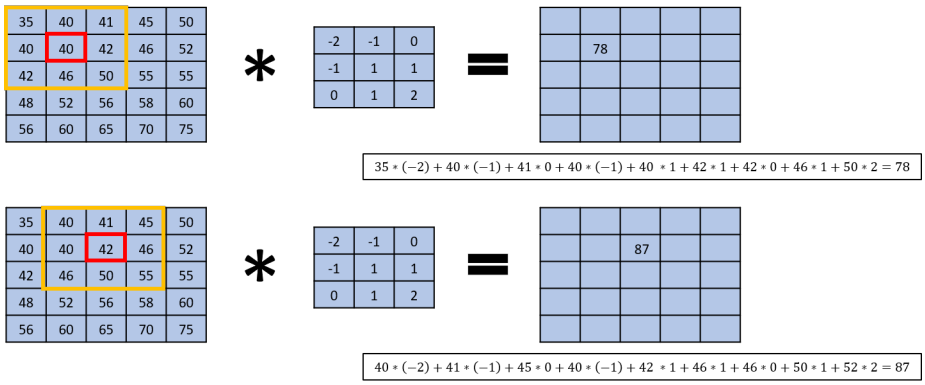


Figura 2.4.10: Ejemplificación de convolución de matrices 1

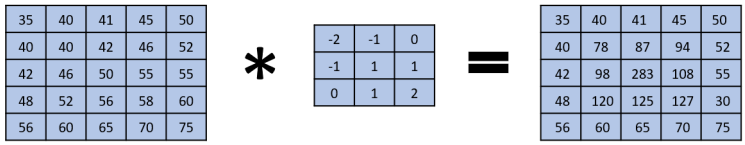


Figura 2.4.11: Ejemplificación de convolución de matrices 2

La convolución se enfoca en el filtrado de imágenes, que consiste en la modificación de ellas haciendo uso del núcleo o kernel se recorre la imagen original, ya sea mejorar o se resaltar algunas de sus características (Bordes, esquinas, etc). Esto tiene la finalidad de obtener información relevante de la misma.

ReLu

ReLu, por su nombre en inglés *Rectified Linear unit* . Se trata de una función de activación iguales a las vistas anteriormente (Sigmoide, lineal, tanh, etc), esta función se le define como:

$$g\,(a) = max(0,\ a) \tag{2.8}$$

Y tiene la representación gráfica que se muestra en la figura 2.4.12:

La idea de utilizar esta función de activación en específico en las RNC es porque cuando procesamos una imagen, cada capa de convolución debe capturar algún patrón en la imagen y pasarla en la siguiente capa de convolución. Los valores negativos no son importantes en el procesamiento de imágenes y, por lo tanto, se establecen en 0.

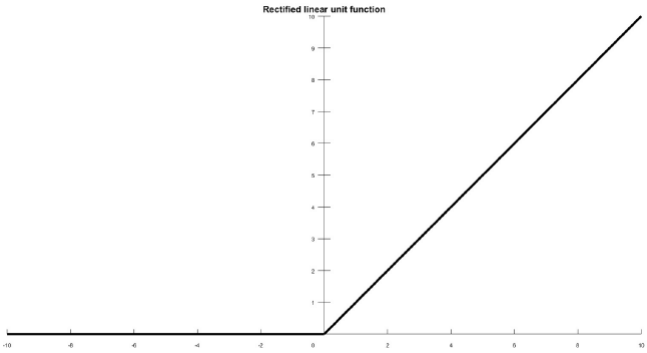


Figura 2.4.12: Grafica de ReLu

Pero los valores positivos después de la convolución deben pasar a la siguiente capa. Es por eso que ReLu se está utilizando como una función de activación. Si utilizamos sigmoide o tanh, la información se pierde ya que ambas funciones modificarán las entradas a un rango muy cerrado.

Pooling

Este es un proceso de discretización basado en muestras. El objetivo es reducir la muestra de una representación de entrada (en nuestro caso una matriz resultante de una imagen), reduciendo su dimensionalidad y permitiendo suposiciones sobre las caracteristicas contenidas en las subregiones agrupadas. Esto reduce el costo computacional, al reducir la dimensión, de digamos una matriz. Para realizar esta discretización, suelen usarse aplicando distintos filtros a una imagen (matriz) de determinado tamaño algunos filtros son el Max Pooling y el Min Pooling, esto quiere decir en el caso del Max Pooling que se utilizará un filtro de nxn que nos dará el número máximo de esa región. Caso contrario del Min Pooling que nos dará el número mínimo de esa región.

Otro parámetro es el “Stride”, el cual marca el número de desplazamientos de píxeles sobre la matriz de entrada. Cuando el stride es 1 entonces movemos los filtros a 1 píxel a la vez. Cuando el stride es 2 entonces movemos los filtros a 2 píxeles a la vez y así sucesivamente.

- Filtros máximos (Max Pooling)

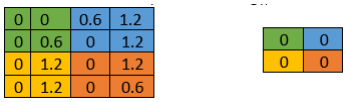


Figura 2.4.13: Representación matricial del Max pooling

- Filtros mínimos (Min Pooling)

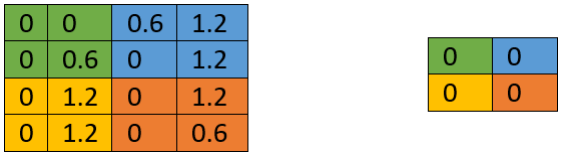


Figura 2.4.14: Representación matricial del Min pooling

2.4.7. Aprendizaje de maquina vs Aprendizaje profundo

Mientras que los algoritmos tradicionales de aprendizaje automático son lineales, los algoritmos de aprendizaje profundo se apilan en una jerarquía de creciente complejidad y abstracción.

Podemos ver el proceso de clasificación de imágenes donde lo anterior se ejemplifica muy bien en el gráfico 2.4.15

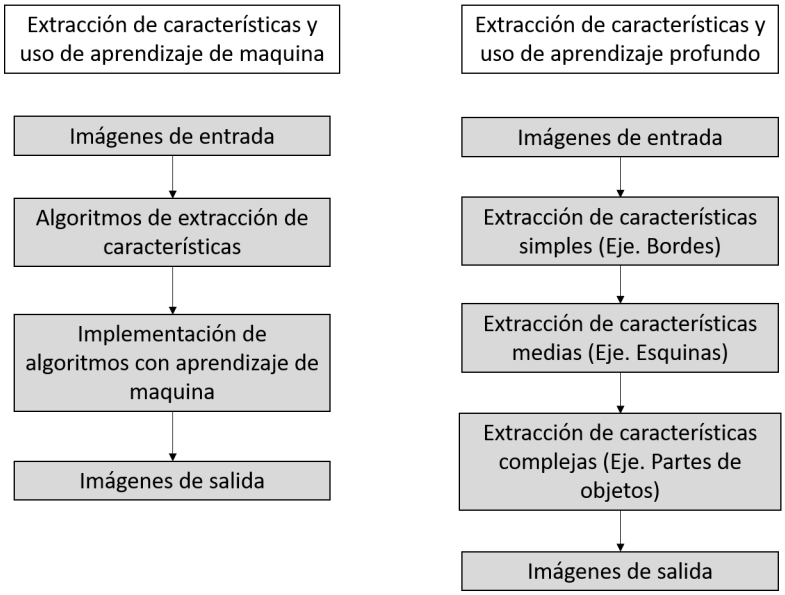


Figura 2.4.15: Comparativa entre algoritmos de aprendizaje profundo y algoritmos de aprendizaje de maquina convencional

Esto quiere decir que en métodos de machine learning tradicionales, es el mismo desarrollador el que tiene que implementar un algoritmo para la extracción de características clave, lo que hace que sea una tarea tediosa y muchas veces es en esta parte donde se lleva la mayor parte del proceso de clasificación. Mientras que la idea de usar técnicas de aprendizaje profundo es la de crear un programa que sea capaz de reconocer esas características por sí mismo, siendo también capaz de aprender a partir de cosas simples como lo son los bordes, esquinas, cambios de color, hasta cosas complejas de una imagen como podrían ser orejas, nariz, ojos, si se estuviera analizando un rostro humano, por ejemplo.

Esto sería parecido a un niño que aprende a diferenciar entre un automóvil y lo que no es un automóvil señalando el mismo con su dedo y siendo corregido por sus padres cuando se equivoca, cada vez que señala, va aprendiendo que características definen a un automóvil.

2.4.8. Visión artificial

Como se ha visto en los temas anteriores, una de las principales filosofías de la computación moderna ha sido la de acercarse cada vez más a las capacidades humanas. Pues si bien desde hace ya bastante tiempo nos han superado en cualquier cosa que involucre operaciones numéricas, otras tareas que el humano considera de lo más comunes aún presentan un reto para las máquinas. Por lo que continuamente se busca dotar con características humanas a las computadoras, entre ellas se encuentran, por ejemplo, los sentidos. Darles capacidades sensoriales a las máquinas no es tarea fácil, ya que entran en juego varias cuestiones como los aparatos mismos que se encargan de recibir

señales y todas las complicaciones que esto conlleva, hasta la misma forma que tenemos de interpretar nuestros propios sentidos.

Aún con estas complicaciones los avances en este campo se han hecho notar y en especial los referentes a un sentido en particular, la visión artificial.

La cual tiene ya muchas aplicaciones en medios sumamente variados, como el control de calidad, hasta la conducción autónoma. Y si bien podríamos decir que la visión artificial se encarga de proporcionarles "ojos" a las máquinas, esta disciplina va más allá de eso, pues según Addison-Wesley [24]:

"La Visión artificial o comprensión de imágenes describe la deducción y propiedades de un mundo tridimensional, posiblemente dinámico, bien a partir de una o varias imágenes bidimensionales de ese mundo."

Esta definición apunta a no solo tratar a la visión artificial como el reconocimiento de formas y figuras geométricas, sino también a las características de estas, como sus materiales o texturas que componen dicho material, una manera más simple de verlo es definiendo a la visión artificial como una disciplina científica que incluye métodos para adquirir, procesar y analizar imágenes del mundo real con el fin de producir información que pueda ser tratada por una máquina.

La visión tanto para humanos o máquinas, consta primordialmente de dos fases: captar una imagen e interpretar dicha imagen. En el caso de las computadoras, la primera fase se lleva a cabo principalmente por cámaras de video y la segunda fase, la de interpretación, se llevaría a cabo por un algoritmo. Aquí es donde el procesamiento digital de imágenes tiene su uso, pues gracias a esta rama de la computación, se pueden llegar a extraer características varias de una imagen para resolver alguna cuestión. Como se mencionaba al inicio de este apartado, se busca dotar a las máquinas de características más cercanas a las humanas o al menos al de entidades "inteligentes" es por ello que hoy no solo las máquinas "ven" sino que también son capaces de aprender, reconocer e interpretar, igual que lo haría cualquier criatura vivía pensante. El ejemplo más grande que tenemos de esto podría venir del campo enfocado en la conducción autónoma de vehículos, pues si pensamos un poco la tarea que involucra conducir esta resulta interesante ya que involucra varios procesos cognitivos complejos.

Capítulo 3

Estado del arte

3.1. Trabajo relacionado

A lo largo de su historia, el ser humano ha tenido la necesidad de retratar su entorno y a sí mismo, tanto que se nos podrían venir a la mente una gran cantidad de obras y artistas que han plasmado momentos clave de la humanidad, aunque cabe decir que este tipo de representaciones usualmente no poseían un carácter práctico, no fue hasta 1824, cuando surgiría el primer procedimiento fotográfico, inventado por Niépce. En el cual las imágenes eran obtenidas con betún de Judea, extendido sobre una placa de plata, luego de un tiempo de exposición de varios días [25].

Ese fue el comienzo de una revolución que cambiaría la forma de ver el mundo para los seres humanos, pues ahora tenían herramientas para detectar fenómenos y estudiarlos de manera más precisa, fue de esta forma que las fotografías se convirtieron en una fuente enorme y muy importante de información para las sociedades modernas.

Años de investigación y con el avance de técnicas para la captura de imágenes junto con el nacimiento y evolución en los sistemas computacionales se llegaron a realizar los primeros trabajos donde se analizaron imágenes por computadora.

Se dice que el inicio de la visión artificial, desde el punto de vista práctico, fue marcado por Lawrence Roberts (1963), el cual creó un programa que podía "ver" una estructura de bloques, analizar su contenido y reproducirla desde otra perspectiva [27], demostrando así a los espectadores que esa información visual que había sido mandada al ordenador por una cámara había sido procesada adecuadamente por él.

Paralelamente, otro gran avance se estaba desarrollando en el campo de la computación, pues desde los años 40's ya nos podíamos topar con conceptos como cibernética, conexionismo, conceptos que actualmente se conocen como, redes neuronales artificiales. Pasando por los trabajos de Warren S. McCulloch y Walter H. Pitts (1943) en donde haciendo uso de una red neuronal crearon en un clasificador binario [22]. Años más tarde, Frank Rosenblatt (1962) con su trabajo, *"Perceptrons and the Theory of Brain Mechanisms"* [28], sentó las bases para el campo de las redes neuronales actuales.

Estos trabajos permitieron que las computadoras fueran capaces de realizar tareas que se pensaban sólo podrían ser realizadas por seres humanos. Un ejemplo de esto fue el primer programa de reconocimiento facial, el cual fue desarrollado por W. W. Bledsoe y L. Browning (1959) donde desarrollaron un sistema que podía clasificar fotos de caras a mano usando lo que se conoce como una tabla RAND, un dispositivo que las personas podrían usar para ingresar coordenadas horizontales y verticales en una cuadrícula usando un lápiz que emitía pulsos electromagnéti-

cos [36]. El sistema podría usarse para registrar manualmente las ubicaciones coordinadas de varias características faciales, incluidos los ojos, la nariz, la línea del cabello y la boca.

En la década de 1970, Goldstein, Harmon y Lesk (1971) le dieron una mayor precisión a un sistema de reconocimiento facial manual. [15] Utilizaron 21 marcadores subjetivos específicos que incluyen el grosor de los labios y el color del cabello para identificar caras automáticamente. Al igual que con el sistema de Bledsoe, la biometría real aún tenía que calcularse manualmente.

Sirovich y Kirby (1986) que comenzaron a aplicar álgebra lineal al problema del reconocimiento facial. [34] Lo que se conoció como el enfoque de Eigenface comenzó como una búsqueda de una representación de baja dimensión de imágenes faciales. Sirovich y Kriby pudieron demostrar que el análisis de características en una colección de imágenes faciales podría formar un conjunto de características básicas. También pudieron demostrar que se requerían menos de cien valores para codificar con precisión una imagen de cara normalizada.

Pero no fue hasta la década de los 90's, gracias al trabajo de Matthew Turk y Alex Pentland (1991) que ampliaron el enfoque de Eigenface al descubrir cómo detectar rostros dentro de las imágenes [35]. Esto condujo a las primeras instancias de reconocimiento facial automático. Su enfoque se vio limitado por factores tecnológicos y ambientales, pero fue un avance significativo para demostrar la viabilidad del reconocimiento facial automático.

A Principios del año 2000, el trabajo de Paul Viola y Michael Jones (2003) presentaron una forma rápida de detección de objetos, usando clasificadores en cascada [38], su algoritmo, tiene la ventaja de poder ser fácilmente implementado por cualquier persona, pues se incluye como parte de la librería de procesamiento digital de imágenes OpenCV.

Además de eso, se hicieron notables avances en otras áreas de la visión artificial, pues S. Zhou, V. Krueger y R. Chellappa (2003) haciendo uso de un algoritmo probabilístico desarrollaron una notable investigación implementando el llamado "algoritmo de condensación" (Condensation algorithm en inglés), para la identificación y seguimiento de rostros de personas a través de video [41]. También es de hacer notar el trabajo de Antitza Dantcheva (2010), el cual consta de un estudio en la llamada "biometría suave" para poder detectar ciertas características de los seres humanos, como el color de la piel, color de cabello y color de ojos, demostrando tener hasta más del 70 % de aciertos respecto a los 3 parámetros anteriormente mencionados [7].

El procesamiento digital de imágenes y las redes neuronales dieron un paso más para la visión artificial, creado una técnica basada en aprendizaje profundo llamada "Redes neuronales convolucionales" y pese a que podemos encontrar un registro de que esta era una idea que ya se venía trabajando desde hace ya tiempo, como nos lo puede demostrar Steve Lawrence (1997) que desarrolló un sistema para el reconocimiento de rostros, basado en redes neuronales convolucionales [18], usar este tipo de técnicas conlleva un alto poder de procesamiento, es por eso que la mayoría de los artículos antes mencionados se basan en técnicas de visión artificial fuera del campo del aprendizaje de máquina.

Hoy en día, gracias al avance en el poder de procesamiento que tienen las computadoras y el auge de hardware especializado como los son las GPUs se han podido crear algoritmos basados en aprendizaje de máquina muy eficaces, un ejemplo de esto es Facenet que aprende mediante un mapeo de imágenes faciales a un espacio euclidiano compacto, además según su artículo [30], presenta una asombrosa precisión de 98.87 %±0.15 en el reconocimiento de rostros.

Como era de esperarse, este tipo de tecnología ha tenido mucho auge en distintos campos, uno de ellos es el uso de aprendizaje profundo en entornos de seguridad. Un ejemplo de esto es el "Sistema de reconocimiento facial FacePRO" de Panasonic Business el cual consta de un complejo sistema de videocámaras y un servidor central el cual analiza las caras y las almacena. Este sistema es capaz de estimar la edad de las personas, género, la frecuencia

con que una persona vista las instalaciones, según su sitio web promocional. Esto es una muestra del gran control que se puede tener sobre un área al implementar este tipo de tecnologías, aunque actualmente, un sistema tan grande y completo, únicamente está disponible para sectores industriales, dejando de lado el uso doméstico. Sin mencionar que también es un servicio restringido a unos cuantos países.

3.2. Análisis de trabajos

En esta investigación se optará por enfocarse en 4 algoritmos específicos. Dos de ellos están basados en aprendizaje de máquina y los dos restantes en aprendizaje profundo. Comenzaremos a estudiar los algoritmos basados en aprendizaje de máquina convencional.

3.2.1. Haar cascade

Detección de objetos mediante clasificadores en cascada basados en funciones de Haar es un método eficaz de detección de objetos propuesto en el documento, “Rapid Object Detection using a Boosted Cascade of Simple Features”. Se trata de un enfoque basado en el aprendizaje automático en el que la función en cascada se entrena a partir de muchas imágenes positivas y negativas. Luego se utiliza para detectar objetos en otras imágenes.

Según sus autores, este algoritmo presenta 3 grandes contribuciones. La primera se trata de una nueva representación de imágenes llamada “Imagen integral” (del inglés “Integral image”), la segunda es un algoritmo de aprendizaje basado en AdaBoost y la tercera contribución es un modelo que combina clasificadores más complejos en “cascada”.

Este algoritmo recorre una imagen en busca de características resaltables de ella, como serían los ojos, distancia que existen entre ellos, nariz, boca, etc. La razón de usar características en el algoritmo es que permite hacer una asociación entre conocimiento aprendido y el análisis de los datos adquiridos. Esto supone un procesamiento más eficiente que el que se tuviera al estar utilizando un análisis basado en píxeles.

Las características de Haar (del inglés Haar-like features), permiten obtener información de una zona concreta dentro de una imagen, mediante una operación aritmética simple: Es por esto que presenta una eficiencia mayor que al tratar con un solo píxel. Para darnos una idea podemos detallar 3 clasificadores Haar

- Característica de borde (edge feature, figura 3.2.1)
- Característica de línea (Line feature, figura 3.2.2)
- Característica de cuatro rectángulos (four-rectangle feature, 3.2.3)

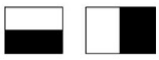


Figura 3.2.1: Características de borde

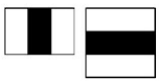


Figura 3.2.2: Características de Línea

Figura 3.2.3: Características de 4 rectángulos

En todos los casos anteriores la obtención del valor de la característica consiste en la resta entre el valor de una o más subzonas dentro de la zona analizada. Es decir, restamos el valor que damos una subzona, mediante la integral sumada, con el de otra subzona.

Para optimizar el cálculo en cada uno de los rectángulos se usa una representación de la imagen llamada "Integral de la imagen" (Del inglés Integral Image). La integral de la imagen respecto a un punto $x,\ y$ consiste en la suma de los pixeles por arriba y a la izquierda de dichos puntos, $x,\ y$ incluidos

$$ii\,(x,\ y) = \sum_{x' \leq\ x,\quad y' \leq y} i(x',\ y') \tag{3.1}$$

La integral de la imagen representa en computación una manera elaborada de obtener valores de forma eficiente. Este método permite el uso de programación dinámica, esto para la obtención de valores dentro de la imagen a través de otros valores calculados previamente. Otro punto importante que tratar es el algoritmo de aprendizaje de maquina AdaBoost.

AdaBoost (Adaptative Boost) es un algoritmo de aprendizaje máquina que consiste en la clasificación de características por medio de un grupo de clasificadores. El algoritmo se basa en la mejor forma de aplicar esos clasificadores para detectar favorablemente algo. En el algoritmo de Viola-Jones el algoritmo AdaBoost es capaz de elegir entre un gran conjunto de filtros, las características de Haar, para seleccionar en cada momento cuál de ellos se ajusta mejor para que se clasifique satisfactoriamente los diferentes elementos que queremos clasificar.

El algoritmo de Viola-Jones utiliza un árbol binario de clasificadores para decidir si una región se trata de una cara o no. Esta cascada está formada por nodos en los que se analizan las diferentes características de las diferentes regiones de la imagen. Cada nodo representa un clasificador obtenido con el algoritmo AdaBoost.

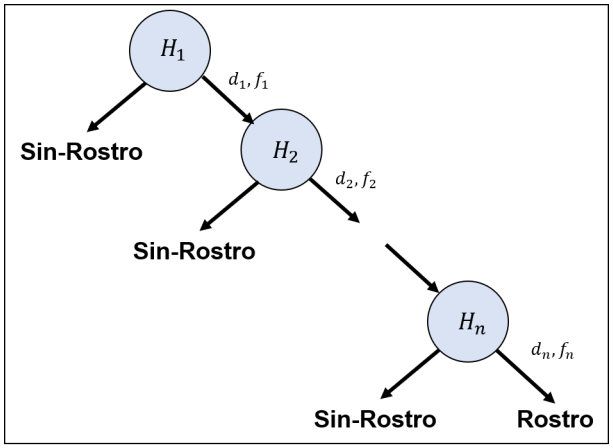


Figura 3.2.4: Árbol binario de clasificación

El proceso de aprendizaje que utiliza el algoritmo de Viola-Jones permite como valor de entrada un error que permitiremos en la clasificación en el primer nodo del árbol.

Utilizando ese valor lo que intenta el algoritmo es crear un clasificador que en el primer nodo permite el error máximo que hemos permitido en la detección y según el árbol se vuelve más profundo el criterio con el que descartamos es mucho más estricto. De esta manera cuando una muestra llega al último nivel del árbol sabemos que ha pasado por todos los niveles de detección, tal como se muestra en la figura 3.2.5.

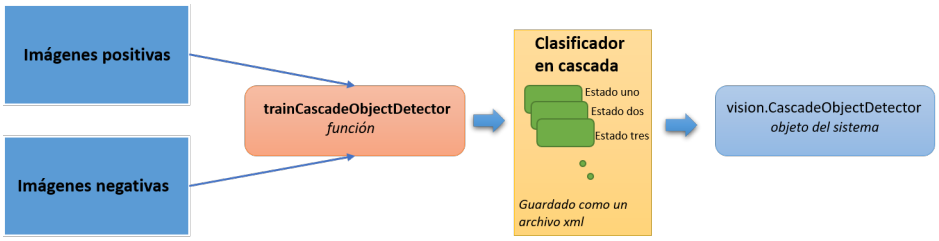


Figura 3.2.5: Esquema general de Haar Cascade.

3.2.2. Histograma de gradientes orientados

En este método de extracción de caracteristicas, presentado por Navneet Dalal y Bill Triggs en su artículo *Histograms of Oriented Gradients for Human Detection* [6] la idea es que la apariencia y la forma local de un objeto, puede ser caracterizado por la manera en que la dirección e intensidad de la iluminación (o color), cambian el contorno realzando así, una forma específica en la imagen. Para saber cómo fue el cambio se hace uso de vectores que describen estos cambios. Estos vectores toman el nombre "vectores gradiente".

Para calcular los gradientes, se toma en cuenta el vector gradiente $\nabla\psi$ de un campo escalar ψ evaluado en un punto ρ, este es un vector que indica la dirección hacia donde el campo varía más rápidamente y el módulo de ese vector indica el ritmo de variación del campo en ese punto evaluado. El gradiente de una función f en coordenadas cartesianas se define de la siguiente forma:

$$\nabla \mathrm{f} = \frac{\delta f}{\delta x} i + \frac{\delta f}{\delta y} j \tag{3.2}$$

El gradiente de una de una imagen capturará el cambio de intensidad o el cambio de color en una de la imagen y la dirección hacia donde cambia. Esto se puede obtener al aplicar un filtro a la imagen, para lograr esto se hace una convolución entre la matriz de pixeles de la imagen y una matriz kernel. En este caso se usarán dos filtros, uno aproxima la derivada en x y el otro en y. De esta forma se obtendrán la derivada de cada píxel en sus respectivos ejes. Es común utilizar el *Filtro de Sobel* y sus matrices se puede ver expresada en la figura 3.2.6.

$$S_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, S_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Figura 3.2.6: Mascaras de convolución del filtro Sobel.

Es así que aplicando una operación de convolución entre cualquiera de estas dos matrices y la matriz de pixeles de la imagen se obtendrá la aproximación de la derivada en el eje x y el eje y, como se podrá apreciar en las siguientes ecuaciones:

$$Gx = S_x * M \tag{3.3}$$

$$Gy = S_y * M \tag{3.4}$$

Donde M es la matriz de pixeles de la imagen a la que se le quiere aplicar el filtro, Gx es la matriz que contiene las aproximaciones de las derivadas en el eje x y Gy para el eje y, un ejemplo practico de la aplicación de estos filtros puede verse en la figura 3.2.7.

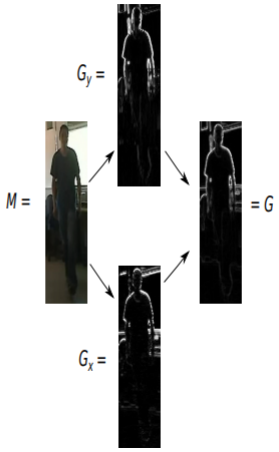


Figura 3.2.7: Ejemplo de la aplicación del filtro Sobel en los ejes x y y.

Y para calcular el gradiente de cada píxel de la imagen se utilizan las ecuaciones siguientes:

$$\theta_{(i,\,j)} = arctan\left(\frac{Gy_{(i,j)}}{Gx_{(i,j)}}\right) \tag{3.5}$$

$$|G_{(i,j)}| = \sqrt{\left(Gx_{(i,j)}\right)^2 + \left(Gy_{(i,j)}\right)^2} \tag{3.6}$$

En resumen, al aplicar este filtro a la imagen de entrada obtenemos como resultado otra imagen que contiene la información de los bordes de los objetos. Esto reduce la cantidad de información que tiene que ser procesada debido a que solo se tomaran en cuenta estos bordes.

Esta síntesis de información se utiliza tanto para hacer el entrenamiento del clasificador, como para analizar la imagen en la que se hará la detección de personas.

Los creadores de este enfoque formaron una máquina de vectores de soporte [37] (Support Vector Machine o SVM) un tipo de algoritmo de aprendizaje automático para la clasificación.

Una máquina de vectores de soporte es un método de aprendizaje automático que, a partir de los datos de entrada, reconoce patrones en estos datos y resuelve problemas de clasificación binaria. El método crea un clasificador

discriminatorio que se define mediante un hiperplano que separa los datos. Estos datos deben estar etiquetados como positivos y negativos para que la salida del método sea un hiperplano optimo que categorizará nuevos datos de entrada.

3.2.3. Faster R-CNN

Esta es una implementación del trabajo *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks* [26] el cual fue entrenado con la base de datos COCO (*Common Objects in Context*) de Microsoft.

El trabajo anteriormente mencionado está basado en *Fast R-CNN* [14] e introduce una técnica conocida como Red de propuestas de región (Del inglés Region Proposal Network RPN) la abreviaremos como RPR. Esta RPR comparte características convolucionales de una imagen completa con la red de detección, lo que permite obtener propuestas de región casi sin costo computacional. Una RPR es una *Fully Convolutional Network* [20], este tipo de redes pueden eficientemente aprender a hacer predicciones en tareas de tipo pixel a pixel como la segmentación semántica (ver la figura 3.2.8) que predice simultáneamente límites de objetos y puntajes de "objectness "en cada posición, la cual es entrenado "end-to-end" generando regiones de alta fidelidad, estás son usadas por Fast R-CNN para detección.

Para tener una mejor compresión de estos conceptos debemos referirnos a Fast R-CNN

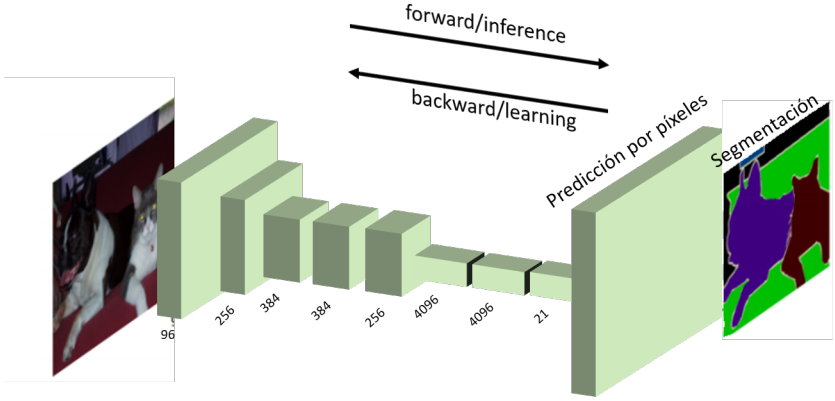


Figura 3.2.8: Ejemplo de una Fully Convolutional Networks.

3.2.4. Detector Fast R-CNN

Podríamos hablar de Fast R-CNN como un descendiente de R-CNN el cual también es un detector de objetos, con el cual, por obvias razones guarda muchas similitudes. Pero también mejora la velocidad de detección, mediante dos argumentos principales:

1. Realizar la extracción de características sobre la imagen antes de proponer regiones, por lo que solo se ejecuta una RNC sobre toda la imagen en lugar de las 2000 regiones superpuestas de 2000 RNC
2. Reemplazar el SVM con una capa softmax, expandiendo así la red neuronal para predicciones en lugar de crear un nuevo modelo

Este detector tiene un modelo parecido al de la figura 3.2.9

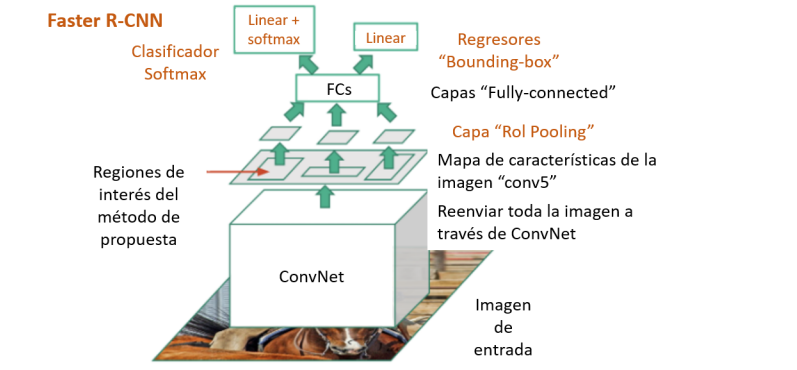


Figura 3.2.9: Ejemplificación del modelo propuesto en Fast R-CNN

Detector Faster R-CNN

El sistema de detección de objetos (Faster R-CNN), está compuesto por dos módulos, el primero es una fully convolutional network que propone regiones (RPR), el segundo módulo es un detector Fast R-CNN que usa esas regiones propuestas. Creando así un solo modelo unificado para la detección de objetos. En este caso la RPR es, utilizando terminología moderna, un *mecanismo de atención* [5] el cual le dice al detector Fast R-CNN "donde mirar"

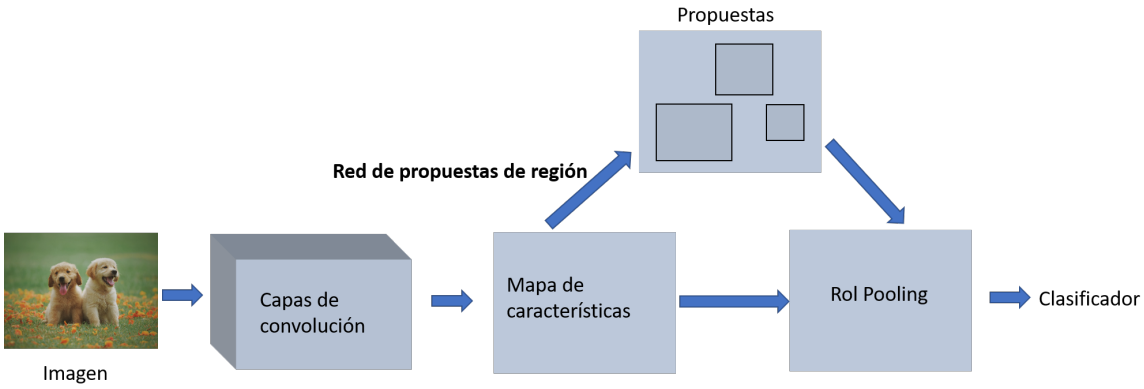


Figura 3.2.10: Faster R-CNN es una sola red unificada para la detección de objetos y el módulo RPR sirve como el mecanismo de atención de esta red unificada

Una red de propuesta de región toma una imagen de cualquier tamaño como entrada y como salida se tiene un conjunto de propuestas rectangulares de objetos cada una con un puntaje "objectness". Esto proceso ha sido modelado con una fully convolutional network con el objetivo de compartir el cálculo con una red para la detección de objetos Fast-R-CNN pues se asume que las dos comparten un conjunto de capas de convolución. En el artículo, los autores mencionan que trabajan con dos modelos para esta RPR el modelo de Zeiler y Fergus [21] (ZF), que tiene 5 capas convolucionales compartibles y el modelo Simonyan y Zisserman [33] (VGG-16), que tiene 13 capas convolucionales compatibles, este proceso se muestra en la figura 3.2.11

El modelo que se usará en este trabajo de investigación será uno que previamente fue entrenado con el conjunto de datos Common Objects in Context [19] (COCO) desarrollado por Microsoft. La razón es por la gran cantidad de imágenes que contiene este conjunto de datos, 80000 imágenes dividías en 80 clases distintas, donde una de estas

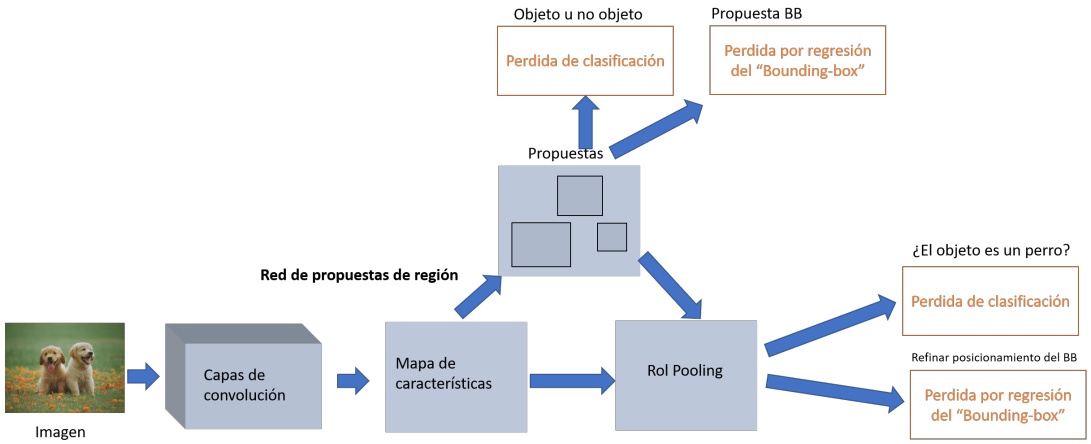


Figura 3.2.11: Ejemplificación del modelo propuesto en Faster R-CNN.

clases es la de figuras humanas. Para ello se entró el modelo haciendo uso de 8 unidades gráficas de procesamiento (GPU por sus siglas en ingles), de esta manera se pudieron implementar un total de 8 RPR (Una por GPU) e implementando el modelo VGG-16 además de 16 Fast R-CNN (2 por GPU) con 240000 iteraciones y una tasa de aprendizaje de 0.003, luego otras 80000 iteraciones con una tasa de aprendizaje de 0.0003.

Un ejemplo del resultado se puede ver en la figura 3.2.12

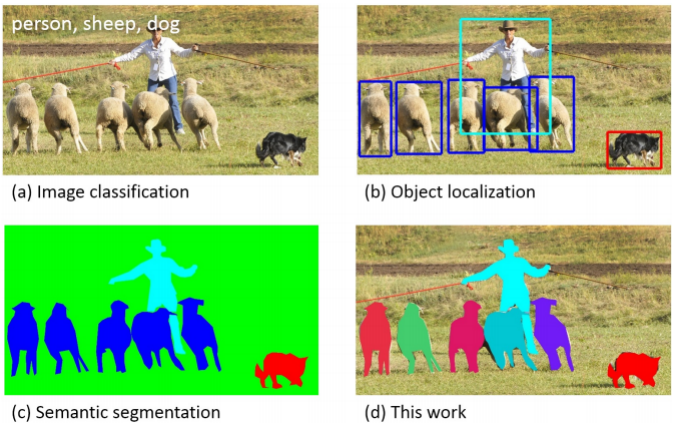


Figura 3.2.12: Resultados de Faster R-CNN.

3.2.5. Redes convolucionales en cascada multitarea

Basado en el trabajo *Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks* [40] MTCNN (de sus siglas en inglés) propone un nuevo framework que consta de tres etapas para realizar la detección de rostros y la detección de puntos de referencia faciales (Ojos, nariz, boca) simultáneamente. En la primera etapa, propondrá varias ventanas candidatas rápidamente a través de una CNN poco profunda. Después de eso, la segunda red refinará las ventanas para rechazar una gran cantidad de ventanas sin caras a través de una CNN más compleja. Finalmente, utiliza una CNN más potente para refinar el resultado y generar posiciones de puntos de referencia faciales, aunque por motivos de practicidad, vamos a rechazar estos últimos y trabajaremos únicamente con la detección de rostros. El proceso puede ejemplificarse en la siguiente figura 3.2.13.

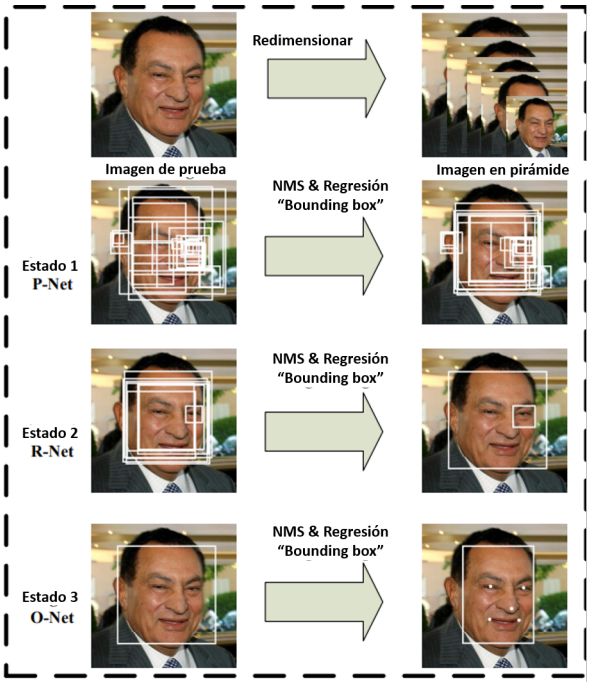


Figura 3.2.13: Los diferentes estados contenidos en MTCNN.

Ahora nos centraremos en explicar los tres diferentes estados mencionados anteriormente por lo que pasa la imagen. Dada una imagen, inicialmente esta es redimensionada a diferentes escalas para construir una pirámide de imagen, que es la entrada del siguiente framework en cascada el cual se divide en tres etapas:

Etapa 1, Se explota completamente red convolucional, llamada Red de propuesta (P-Net), para obtener los frames candidatos y sus vectores de regresión para un cuadro delimitador de manera similar al trabajo realizado en *Multi-view Face Detection Using Deep Convolutional Neural Networks* [9], Luego se usan los vectores de regresión estimados del cuadro delimitador para estimar a los frames candidatos. Luego de eso se utiliza una supresión no-máxima o SNM (non-maximum suppression o NMS) para fusionar candidatos altamente superpuestos.

Etapa 2: Todos los frames candidatos son enviados a otra CNN, llamada Refine Network (R-Net), que rechaza aún más una gran cantidad de candidatos falsos, realiza la calibración con regresión de cuadro delimitador y fusión de candidatos SNM.

Etapa 3: Esta etapa es similar a la segunda etapa, pero en esta etapa el objetivo es describir la cara con más detalles. En particular, la red generará cinco posiciones de puntos de referencia faciales.

Capítulo 4

Algoritmo para el reconocimiento de rostros

4.1. Introducción

De acuerdo con la fase de experimentación, se optará por algoritmos que presenten una solución ideal de acuerdo con el problema presentado, nos guiaremos por los resultados de los experimentos para tener una base sólida para argumentar el uso de un algoritmo en específico sobre otro. Esto, además, tiene como objetivo encontrar el balance entre cuadros por segundo y resolución a la cual es conveniente trabajar; aunado a esto, se pretende dar con una concordancia entre la complejidad (De acuerdo al marco teórico) con los resultados experimentales, pues recordemos que estaremos trabajando con dos grupos de algoritmos (Aprendizaje de máquina convencional y aprendizaje profundo), por lo tanto, podemos hacer estimaciones como qué tipo de algoritmos tendrán un mayor tiempo de procesamiento y así como precisión más alta.

Como se ha explicado con anterioridad el problema planteado se basa principalmente en un tema de seguridad, pues como se explicó en el apartado referente al planteamiento del problema, los robos a casa habitación son un grave problema que vive la sociedad actualmente, por esto mismo dotar de herramientas para combatir este tipo de crímenes y que además se presente una solución que no requiera una inversión muy alta se hace necesario. Es por esto mismo que algo característico de la solución planteada es la posibilidad de implementar la solución con equipos de cómputo relativamente modestos y que además presenten una buena precisión en la detección de personas.

Para lograr esto, se busca aplicar los 4 algoritmos estudiados con anterioridad. A grandes rasgos podemos separar en dos grupos estos algoritmos.

- Aprendizaje de máquina convencional
    - Haar cascade
    - Histograma de gradientes orientados (HOG)
- Aprendizaje profundo
    - Faster R-CNN (Pre-entrenada con el conjunto de datos COCO)

- Multi-task Cascaded Convolutional Networks (MTCNN)

A su vez tenemos dos tareas principales que resolver, las cuales son la detección de siluetas y la detección de rostros. Estas tareas serán tratadas por los algoritmos de la siguiente manera:

- Detección de siluetas
  - Haar cascade
  - Histograma de gradientes orientados (HOG)
  - Faster R-CNN
- Detección de rostros
  - Haar cascade
  - Histograma de gradientes orientados (HOG)
  - Multi-task Cascaded Convolutional Networks (MTCNN)

La razón de implementar los algoritmos de aprendizaje de maquina convencional tanto en detección de siluetas como de rostros es porque estos tienen implementaciones para ambos casos, mientras que Faster R-CNN como MTCNN únicamente funcionan para sus tareas designadas, la detección de siluetas para el caso de Faster R-CNN y la de rostros para MTCNN.

Estos algoritmos buscan ser implementados en ambientes similares a los que muestran las cámaras de vigilancia.

4.2. Esquema de solución

A continuación, se explicará cómo es que se trabaja para solucionar este problema por medio de un esquema de solución representado por la figura 4.2.1.

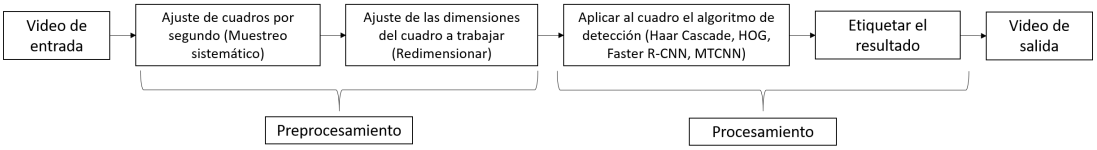


Figura 4.2.1: Esquema general de solución

Como se puede observar en la figura anterior, se ha optado por dividir la tarea en dos grandes segmentos, el primero el preprocesamiento, donde se realizan los ajustes de cuadros por segundo y dimensiones, es decir la parte donde el algoritmo de detección no se utiliza, pero sí se trata con la entrada que tendrá este, a su vez estos ajustes afectaran el rendimiento del algoritmo de detección. En la segunda etapa es donde se aplica el algoritmo de detección al cuadro seleccionado, también donde se valida la salida del mismo (Encontró o no encontró algo) para poder marcar (o no) la silueta o el rostro detectado en el cuadro en cuestión.

Cabe señalar que estos dos grandes bloques de trabajo se pueden realizar por separado, es decir se pueden tomar un conjunto de videos por separado y aplicarles el preprocesamiento, para luego únicamente pasar al procesamiento los videos con los ajustes en dimensiones y cuadros por segundo deseados. En ese caso se tiene algo parecido a la figura siguiente 4.2.2 .

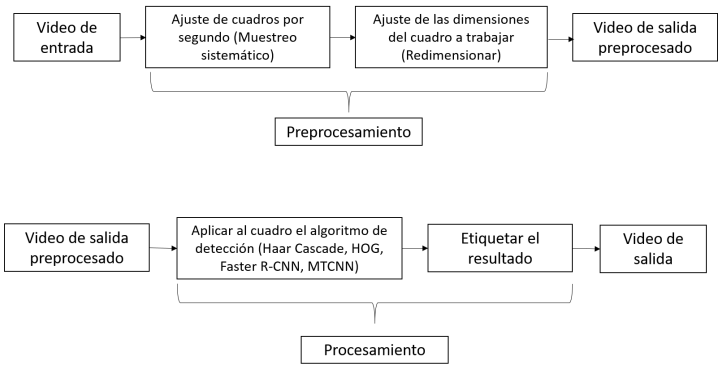


Figura 4.2.2: Separación de los bloques de trabajo

Cabe mencionar que los experimentos presentados en esta investigación se harán siguiendo el esquema "Procesamiento", tal como se muestra en la figura 4.2.2. Pues se pretende enfocar en el rendimiento especifico de los algoritmos sin tener que pasar por el proceso de redimensionar imágenes o disminuir los fotogramas por segundo.

4.3. Aplicación de pasos de la solución

Para seguir el modelo de solución e implementar los pasos acordados se hará uso del lenguaje de programación Python en su versión 3.7.4, pues este incluye las librerías necesarias para trabajar con los algoritmos propuestos, concretamente la distribución de Anaconda por la facilidad que tiene para crear diferentes ambientes de trabajo.

Las paqueterías de Python que usarán de manera primordial en este proyecto son las siguientes:

- OpenCv
- TensorFlow
- NumPy
- Dblib
- Time

Como se explicó anteriormente, se trabajará de manera separada entre el preprocesamiento y el procesamiento de un video. Por lo que pasaremos a definir en qué consiste cada uno de los módulos que componen los dos bloques de trabajo, comenzando por el preprocesamiento.

Ajuste de cuadros por segundo

Como se definió desde el principio, esta es una métrica que va a definir directamente el rendimiento en general de la solución. Usualmente, el tipo de cámaras dedicas a la videovigilancia operan arriba de los 24 fotogramas por segundo hasta los 60 FPS en casos particulares, como se definió en el marco teórico, esta cantidad de FPS puede ser reducida pues con menos FPS al menos de manera teórica, ningún ser humano escapará de ser captado por la cámara (Esto puede cambiar en función de la distancia a la que se sitúe la cámara). Por esto mismo, se necesita hacer un ajuste de la cantidad de fotogramas por segundo que estaremos implementando. Para ello, se ha implementado un algoritmo que de alguna manera "filtre" determinado número de fotogramas.

Imaginemos un video que este grabado a 30 fotogramas por segundo y tenga una duración de 10 segundos. En total se tienen 300, pero si se requiere de pasar de 30 fotogramas a 6 fotogramas por segundo se contabiliza un total de 60 fotogramas, pero no podemos simplemente quitar los primero o los últimos fotogramas del video pues esto nos dejaría con video con algunas escenas completamente perdidas (Véase el final o el inicio), lo que queremos es eliminar fotogramas de tal manera que no se pierda la continuidad del video, por lo que tenemos que ir quitando fotogramas intermedios

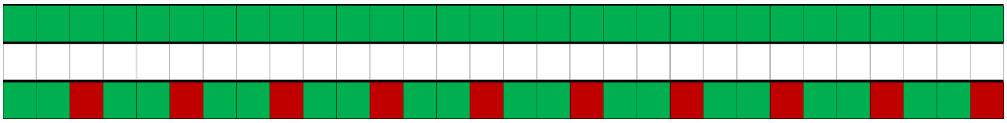


Figura 4.3.1: Eliminación de fotogramas de un video

En la figura 4.3.1 podemos ver cómo sería un ejemplo del proceso de eliminación de fotogramas, tomando como referencia el video a 30 fotogramas por segundo, se ejemplifica en los recuadros verdes los 30 fotogramas, y con los recuadros rojos los fotogramas eliminados. Pasando de tener 30 fotogramas en un segundo a 20 fotogramas.

Algo importante que se considera pertinente mencionar, es el tiempo final del video. Pues no por eliminar fotogramas contenidos en un segundo, significa que se va a reducir el tiempo de este. Lo único que se busca es reducir la "fluidez" al reducir el número de cuadros por segundos contenidos en él.

Esto se puede apreciar como un muestreo sistemático [3], donde dividimos nuestro marco muestral, en este caso la cantidad de frames que componen un video, en otro video con menor número de muestras (Frames), dando como resultado un video con menos cuadros por segundo.

Capítulo 5

Fase de experimentación

En este apartado se hará uso de la propuesta de solución aplicada a distintos conjuntos de datos para poder ver los resultados que estos tendrán en función de los algoritmos descritos en los capítulos anteriores. Para esto se ha investigado una amplia gama de conjuntos de datos, todos en video, los cuales son sumamente variados (Distintos fotogramas por segundo y distintas resoluciones), por lo que se espera tener un panorama amplio de aplicación de los algoritmos, así como tener una idea general de cómo se comportan los mismos en distintos tipos de ambientes.

Aún con esto dicho, se realizarán pruebas donde todos los algoritmos estarían bajo las mismas condiciones de resolución, fotogramas por segundo y tipo de archivo, de esta manera se tendrán a la los algoritmos en igual de condiciones.

5.1. Metodología de las bases de datos

Como se mencionó anteriormente, lo que se busca es tener variedad en cuanto a los diferentes videos, cabe mencionar que todos ellos son de libre acceso. Por lo que la replicación de estos experimentos será de manera más sencilla. A continuación, se detallarán los conjuntos de datos, ver figura 5.1.1.

Como se puede observar se dispone de un amplio conjunto de datos con los cuales experimentar. Por motivos de tiempo será imposible procesar cada uno de los videos en sus diferentes configuraciones. Por lo que se hace indispensable seleccionar conjuntos de datos que estén más apegados a los escenarios presentados por cámaras de videovigilancia.

5.2. Metodología de experimentación

A continuación, se detallarán los pasos para la experimentación y así como los conjuntos de datos seleccionados los cuales son:

- BEHAVE [17]
- CAVIAR [10]
- CamNet [29]

Nombre	Número de videos	Tamaño total	Formato de video	Dimensiones del video (pixeles)	FPS	Descripción
300VW Dataset 2015_12_14 [10]	50	1.85 GB	.avi	1280 x 720	25	Videos de programas de televisión donde se enfoca un solo rostro
Aerial Gait dataset [11]	17	3.39 GB	.mp4	1920 x 1080	25	Videos donde una sola persona camina, realizando una ruta predeterminada
CamNet [24]	8	1.54	.avi	704 x 480	20	Videos de corredores grabando varias personas
CAVIAR [8]	26	274 MB	.mpg	284 x 288	25	Videos de varias personas realizando actividades predeterminadas
Ixmas [37]	1800	1.16 GB	.avi	48 x 64	25	Videos de una sola persona realizando poses específicas
KitWare data [2]	93	5.39 GB	.mp4	1028 x 720	30	Videos tomados desde dos locaciones diferentes desde un punto alto
Dataset atomic pair actions [29]	90	253 MB	.wmv	650 x 480	30	Videos de cámaras de vigilancia registrando 3 diferentes tipos de movimiento
BEHAVE [12]	4	300 MB	.wmv	640 x 480	25	Videos de personas realizando poses y actividades varias en dos locaciones distintas

Figura 5.1.1: Detalles de los conjuntos de datos

Mientras que los elementos que no utilizamos, pero se pueden considerar para trabajos futuros son:

- 300VW [11]
- Aerial [12]
- Ixmas [39]
- KitWare data [2]
- Atomic Pair Actions [31]

Esto se debe a que en la base de datos BEHAVE tendremos una cámara que está posicionada para exteriores capturando a una distancia considerable, mientras que CamNet y CAVIAR tienen una cámara enfocada a interiores y exteriores reducidos. Esto se presta a tener conocimiento de una distancia media a la cual los algoritmos, tanto de reconocimiento de siluetas como de rostros dejan de trabajar correctamente.

Los experimentos se llevaron a cabo de tal manera, que se pudieran procesar todos los videos de un determinado conjunto de datos y de esta manera registrar los siguientes datos:

1. Tiempo de procesado por video
2. FPS promedio por video
3. Conteo de activaciones del algoritmo por video

5.2.1. Tiempo de procesamiento por video

Este valor está dado por el tiempo en que un algoritmo tarda en procesar todos los cuadros de un video, es decir que, si un video está configurado a 20 cuadros por segundo y a su vez tiene una duración de 10 segundos, quiere decir que el algoritmo tendrá que procesar un total de 200 cuadros. Este tiempo guarda una relación de proporción con los FPS de salida de cada video.

5.2.2. FPS promedio por video

Este valor es dado por el tiempo de procesamiento por video dividido por el número total de cuadros que tenga un video, de tal manera que podremos obtener la siguiente fórmula.

Con esto se puede reafirmar que la relación que se tiene entre el tiempo de procesamiento y el de cuadros por segundo totales es que los cuadros por segundo totales son inversamente proporcionales al tiempo de procesamiento, pues entre más grande se haga este tiempo de procesamiento, menos FPS se tendrán en la salida de video.

Una consideración que debe tenerse en cuenta es que no es lo mismo tener como salida n FPS en un video procesado a 4 cuadros por segundo que uno procesado a 10 cuadros por segundo. Pues suponiendo que se trate del mismo video con una duración de 15 segundos, la cantidad de cuadros totales que se tienen en cada caso serían las siguientes:
$4FPS * 15\ segundos = 60\ Cuadrostotales$
$10FPS * 15\ segundos = 150\ Cuadrostotales$

Por lo tanto, suponiendo que en el primer caso con un algoritmo en específico se obtenga un $Tp = 3\ segundos$, mientras que el segundo obtenga un $Tp = 7{,}5\ segundos$, siguiendo la fórmula anterior

$$FPS = \frac{60}{3} = 20 \tag{5.1}$$

$$FPS = \frac{150}{7{,}5}\ 20 \tag{5.2}$$

Ambos videos tienen salidas de 20 FPS, pero el tiempo de procesamiento de cada uno es diferente.

Es por esto, que la principal métrica que consideraremos será la del tiempo.

Otro punto importante en la fase de experimentación es que, si un determinado vídeo digamos a 4 FPS, de como resultado 23 FPS en sus FPS promedio, el video que se guardará seguirá estando a 4 FPS

5.2.3. Conteo de activación del algoritmo por video

La forma en la que trabajan estos algoritmos (ya sean los orientados a rostros o a siluetas) es la de encontrar una forma en específica y así mandar las coordenadas (ejes x, y) dentro de la imagen, de esta manera se obtienen los puntos que formarían un cuadrilátero en la imagen (ver la figura 5.2.1), por lo que la salida de la detección es en realidad un arreglo con un conjunto de coordenadas.

Figura 5.2.1: Ejemplo de la salida de un algoritmo de detección de siluetas

El conteo de activación, se encarga de contar el número de veces que estos arreglos, no resultan vacíos.

Podemos decir que las variables directamente proporcionales al tiempo de procesamiento de un video son, la resolución y la cantidad de cuadros por segundo que contengan y por lo tanto las primeras pruebas se harán precisamente comparando los resultados de procesar el mismo video con cuadros por segundo y resoluciones distintas.

5.2.4. Ambiente de trabajo

Con el fin de tener versatilidad al momento de probar diferentes algoritmos, se decidió por la utilización de Jupyter con el lenguaje propósito general Python, en su versión 3.7.

Trabajando en una computadora con las siguientes características:

- Procesador: Ryzen 5 3600x a 3.6 GHz
- Memoria RAM: 16 GB a 3200 MHz
- Almacenamiento: 960 GB M.2 nvme

5.3. Experimentación con detección de personas

5.3.1. Análisis del tiempo de procesamiento en función de la resolución

Para este análisis, vamos a estudiar el resultado de usar los métodos de reconocimiento de siluetas con las técnicas basadas en aprendizaje de máquina convencional, con los dos algoritmos propuestos con anterioridad, Haar cascade y el histograma de gradientes orientados que llamaremos simplemente HOG (Por sus siglas en inglés Histogram Of Oriented Gradients). Al cual le pasaremos un conjunto de datos en específico.

Las resoluciones que iremos variando son las siguientes:

- Resolución original
- FHD (1920 x 1080 píxeles)
- HD (1280×720 píxeles)
- 352 x 288 píxeles
- 176 x 144 píxeles

La resolución original del video (Varía dependiendo del conjunto de datos)

Las siguientes pruebas se realizarán con los FPS originales de cada conjunto de datos, estos están marcados dentro de un paréntesis luego del nombre de la base de datos.

Pruebas con CAVIAR (25 FPS)

En las figuras 5.3.1 y 5.3.2 se muestra un gráfico en función del tiempo en función de la resolución de los vídeos implementando el algoritmo Haar cascade y HOG.

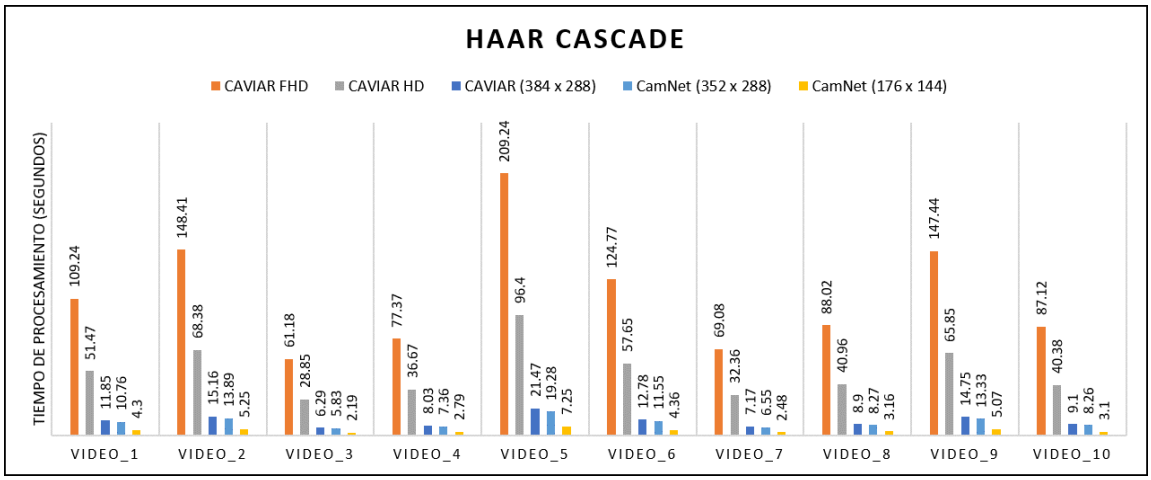


Figura 5.3.1: Gráfica de Haar Cascade en función del tiempo (25 FPS)

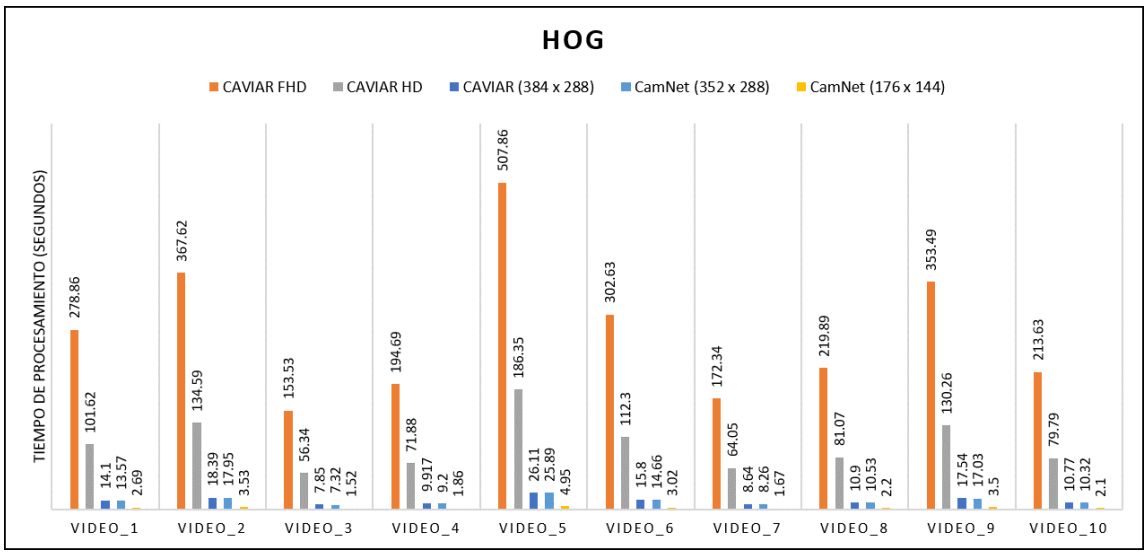


Figura 5.3.2: Gráfica de HOG en función del tiempo (25 FPS)

Pruebas con CamNet (20 FPS)

En las figuras 5.3.3 y 5.3.4 se muestra un gráfico en función del tiempo en función de la resolución de los vídeos implementando el algoritmo Haar cascade y HOG.

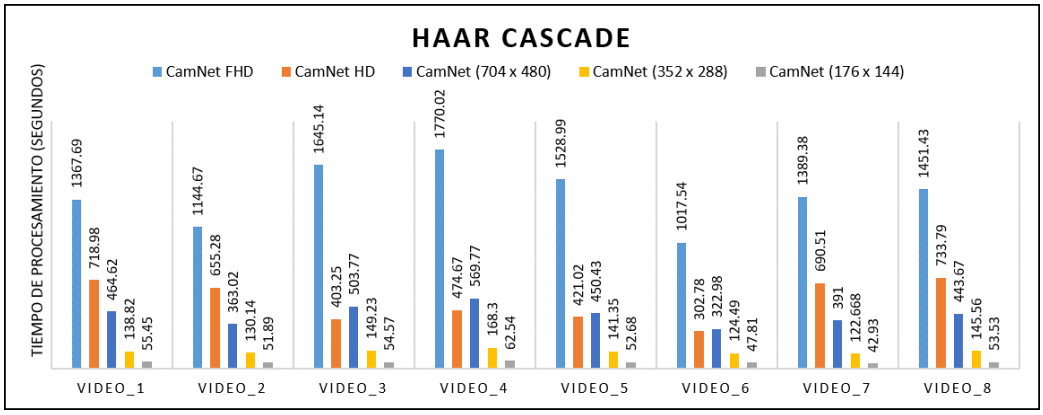


Figura 5.3.3: Gráfica de Haar Cascade en función del tiempo (20 FPS)

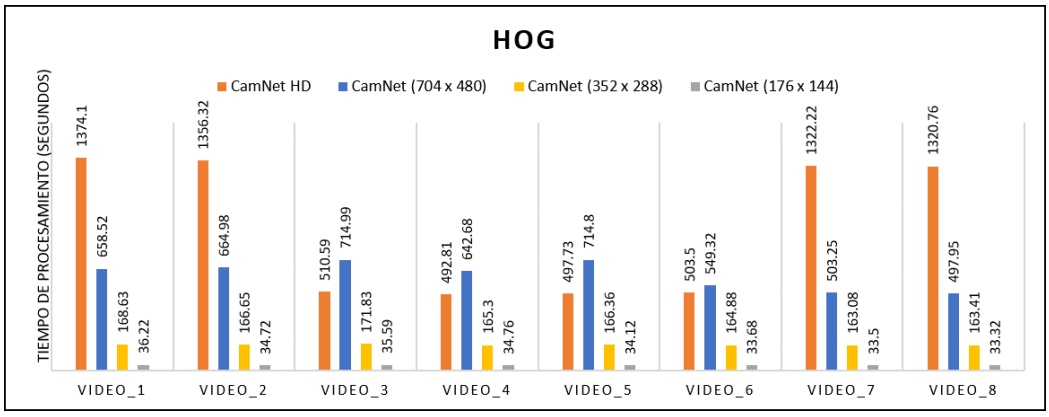


Figura 5.3.4: Gráfica de HOG en función del tiempo (20 FPS)

Con lo visto anteriormente, podemos decir lo siguiente:

1. Como era de esperarse, a mayor resolución, mayor tiempo de procesamientoz
2. Trabajar con videos en resoluciones HD en adelante, se hace prácticamente inviable en términos del tiempo requerido, este problema se acentúa si se pretende trabajar video en tiempo real.
3. HOG tiende a requerir casi el doble tiempo de procesamiento que Haar cascade en resoluciones arriba del HD y esta diferencia comienza a disminuir a medida que las resoluciones decrecen hasta llegar al caso de la resolución (176 x 144) donde Haar Cascade supera en tiempo de procesamiento a HOG.

5.3.2. Análisis del tiempo de procesamiento en función de los cuadros por segundo

Ahora se procederá a analizar el comportamiento de los distintos conjuntos de datos al reducirse los FPS. Los FPS a los que se trabajarán serán los siguientes, por las cuestiones vistas en el marco teórico referentes a la distancia que un ser humano es capaz de recorrer en un segundo, FPS originales (Varían dependiendo del conjunto de datos)

- 5 FPS
- 4 FPS
- 3 FPS

Y se optará por trabajar con 2 resoluciones, la original y la resolución de 352 x 288.

Pruebas Caviar

En las figuras 5.3.5 y 5.3.6 se muestra una gráfica del tiempo de procesamiento en función de los cuadros por segundo, considerando también dos resoluciones diferentes, implementando los algoritmos de Haar cascade y HOG con la base de datos Caviar.

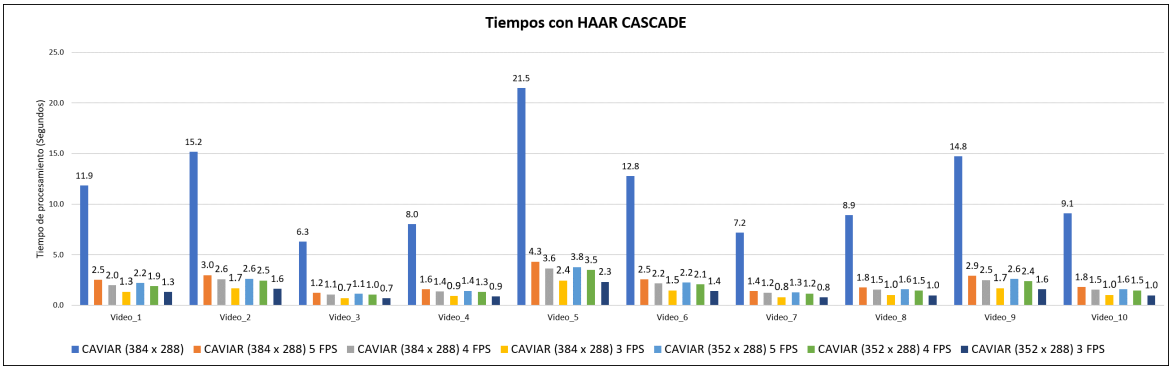


Figura 5.3.5: Gráfica de Haar Cascade en función del tiempo (25 FPS)

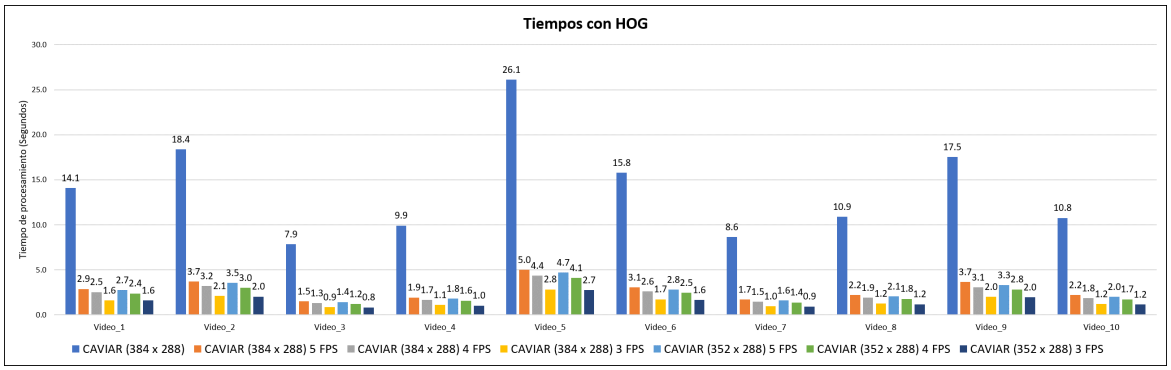


Figura 5.3.6: Gráfica de HOG en función del tiempo (25 FPS)

Pruebas CamNet

En las figuras 5.3.7 y 5.3.8 se muestra una gráfica del tiempo de procesamiento en función de los cuadros por segundo, considerando también dos resoluciones diferentes, implementando los algoritmos de Haar cascade y HOG con la base de datos CamNet.

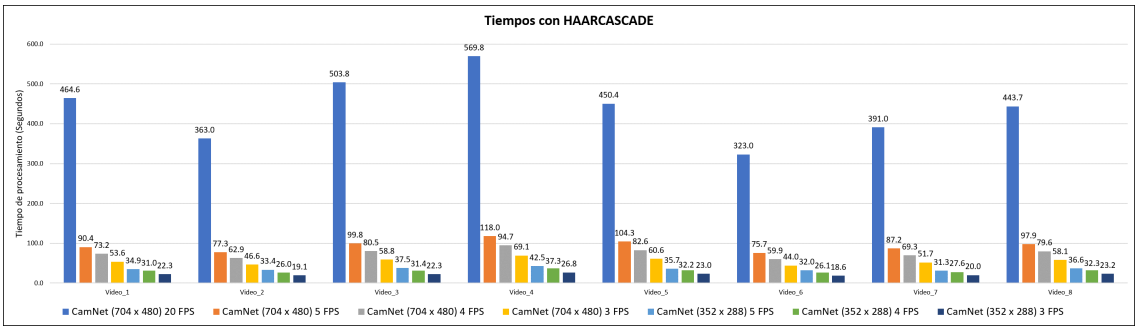


Figura 5.3.7: Gráfica de Haar Cascade en función del tiempo (25 FPS)

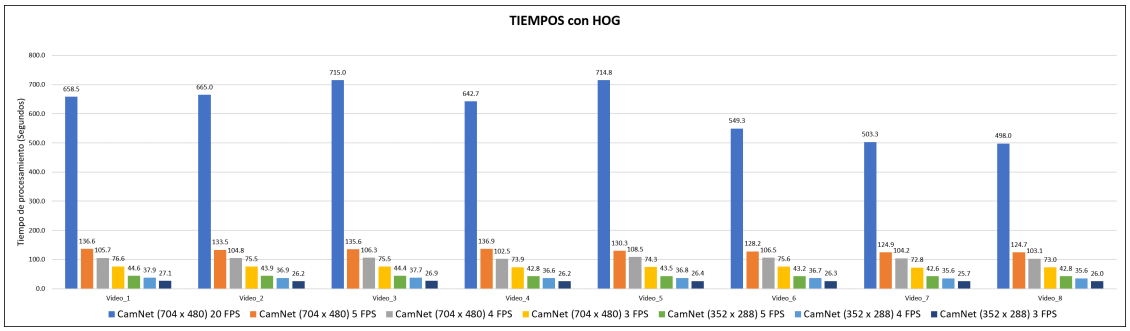


Figura 5.3.8: Gráfica de HOG en función del tiempo (25 FPS)

Pruebas BEHAVE

En las figuras 5.3.9 y 5.3.10 se muestra una gráfica del tiempo de procesamiento en función de los cuadros por segundo, considerando también dos resoluciones diferentes, implementando los algoritmos de Haar cascade y HOG con la base de datos BEHAVE.

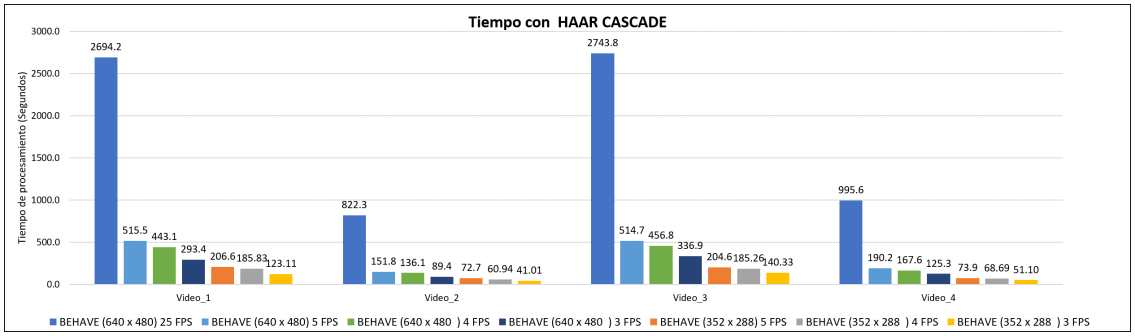


Figura 5.3.9: Gráfica de Haar Cascade (BEHAVE) en función del tiempo (25 FPS)

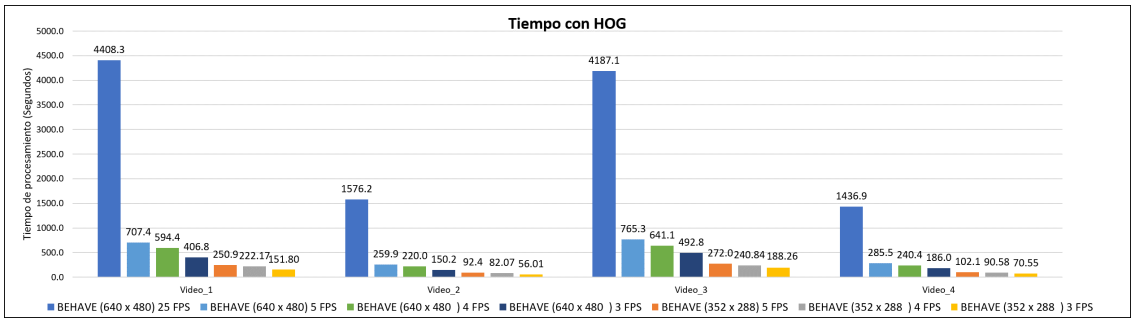


Figura 5.3.10: Gráfica de HOG (BEHAVE) en función del tiempo (25 FPS)

En conclusión, también se hace indispensable considerar la idea de tener pocos FPS y una resolución reducida para poder implementar un algoritmo que trabaje con un hardware no tan costoso.

5.3.3. Personas detectadas

Se hace indispensable comprobar la precisión de cada algoritmo al momento de detectar personas, para esto se realizará una serie de pruebas donde se hará un conteo de la cantidad de personas que aparezcan en un video, luego se hará el mismo conteo esta vez teniendo en cuenta únicamente las siluetas detectadas por un algoritmo, nuevamente, estas pruebas se realizarán con los algoritmos basados en aprendizaje de máquina y los conjuntos de datos previamente utilizados.

Hay que mencionar además que se utilizarán los mismos parámetros de resolución anteriores:

- Resolución original
- 352 x 288 pixeles

Las graficas que se verán a continuación (Figura 5.3.11 hasta la figura 5.3.22) representan cada video dentro de su conjunto de datos.

Pruebas BEHAVE

En la figura 5.3.11 hasta la figura 5.3.14 se muestran gráficas de las personas detectadas por algoritmo (Haar cascade y HOG) en función de cuadros por segundo y resoluciones distintas. Además, en color verde se observa el total de personas en dicho video. Todos los videos son correspondientes a la base de datos BEHAVE.

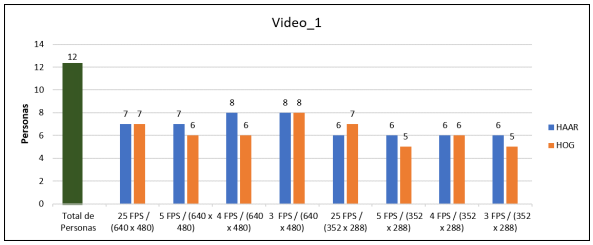


Figura 5.3.11: Gráfica de siluetas detectadas en el conjunto de datos BEHAVE

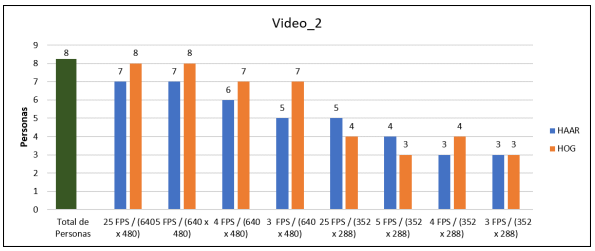


Figura 5.3.12: Gráfica de siluetas detectadas en el conjunto de datos BEHAVE

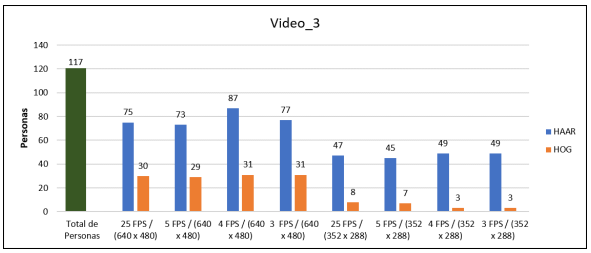


Figura 5.3.13: Gráfica de siluetas detectadas en el conjunto de datos BEHAVE

Pruebas CamNet

En la figura 5.3.15 hasta la figura 5.3.22 se muestran gráficas de las personas detectadas por algoritmo (Haar cascade y HOG) en función de cuadros por segundo y resoluciones distintas. Además, en color verde se observa el total de personas en dicho video. Todos los videos son correspondientes a la base de datos CamNet.

De estas gráficas se pueden sacar varias conclusiones

- No se tiene una diferencia muy signif icativa en el cambio de resoluciones para Haar Cascade o HOG, aunque este último tiene una pérdida de precisión mayor con respecto a Haar Cascade
- HOG tiene una pérdida de precisión muy grande al tratar con videos donde las personas se encuentren en áreas más alejadas, es decir, en ambientes de exteriores

Esto reafirma la idea de trabajar con videos a baja resolución y a bajos FPS, pues por lo visto en las gráficas, su impacto en la precisión no es muy considerable. Este comportamiento, está aún más presente en Haar Cascade.

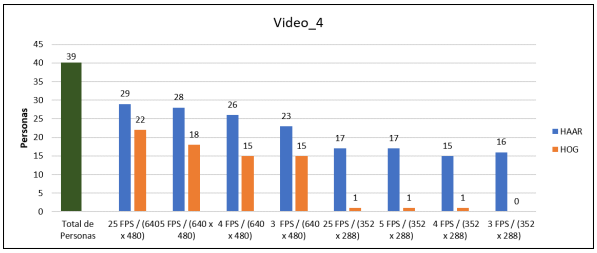


Figura 5.3.14: Gráfica de siluetas detectadas en el conjunto de datos BEHAVE

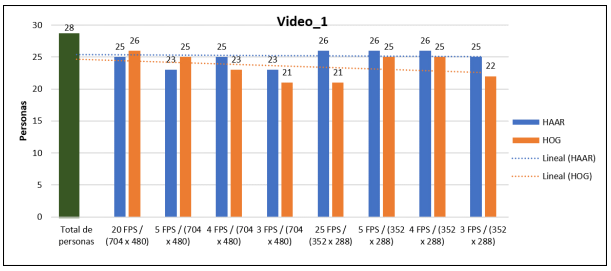


Figura 5.3.15: Gráfica de siluetas detectadas en el conjunto de datos CamNet

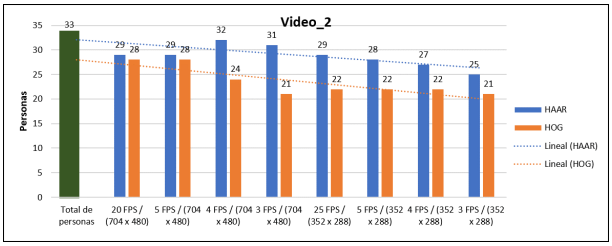


Figura 5.3.16: Gráfica de siluetas detectadas en el conjunto de datos CamNet

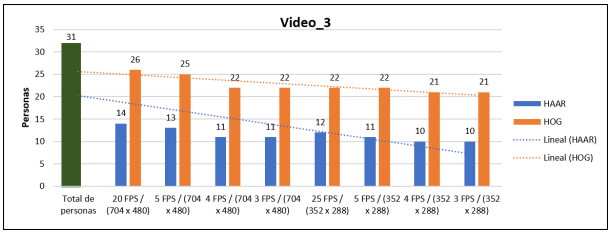


Figura 5.3.17: Gráfica de siluetas detectadas en el conjunto de datos CamNet

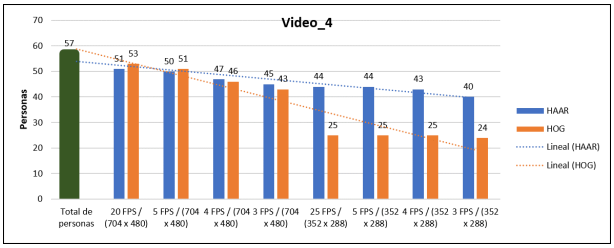


Figura 5.3.18: Gráfica de siluetas detectadas en el conjunto de datos CamNet

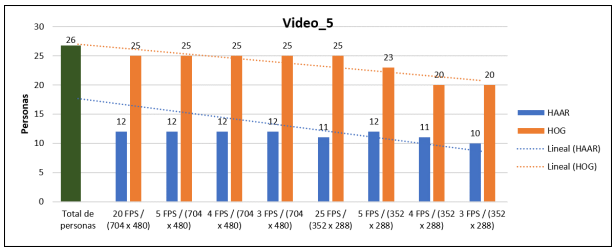


Figura 5.3.19: Gráfica de siluetas detectadas en el conjunto de datos CamNet

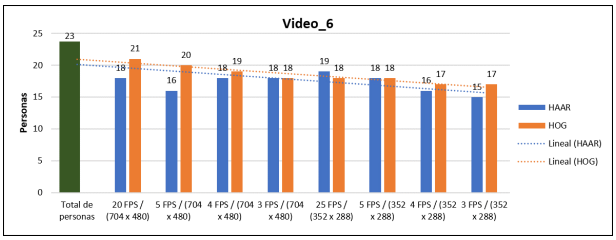


Figura 5.3.20: Gráfica de siluetas detectadas en el conjunto de datos CamNet

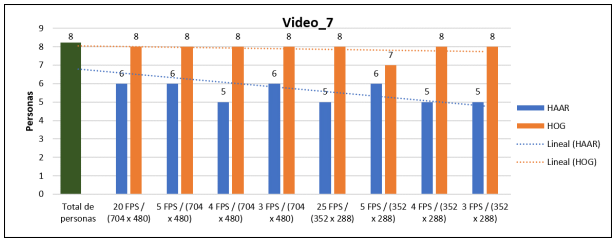


Figura 5.3.21: Gráfica de siluetas detectadas en el conjunto de datos CamNet

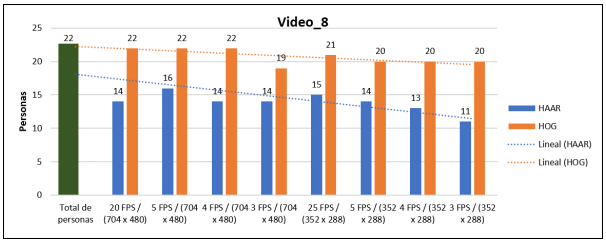


Figura 5.3.22: Gráfica de siluetas detectadas en el conjunto de datos CamNet

5.3.4. Verdaderos-Positivos, Falsos-Positivos y Falsos-Negativos

Lo siguiente a analizar es qué tan fiables al momento de detectar “algo” son cada uno de los diferentes algoritmos, esto es porque se puede prestar el escenario donde el algoritmo se active, pero se active de forma incorrecta y para esto se llevará a el más bajo nivel de granularidad la observación en un video, esto se refiere a estudiarlo cuadro a cuadro, para esto se seleccionaron 8 muestras de videos de los diferentes conjuntos de datos con una duración aproximada de 3 segundos. Por la evidencia vista anteriormente, se eligió estudiar el comportamiento en específico sobre videos a 4 cuadros por segundo en una resolución de 352 x 288 pixeles.

Los casos a estudiar serán los siguientes:

1. Verdaderos-Positivos: Caso donde se tiene a una persona y es detectada

Figura 5.3.23: Ejemplo de un caso Verdadero-Positivo

2. Falsos-Positivos: Caso donde se detecta “algo” como una persona, sin ser una persona

Figura 5.3.24: Ejemplo de un caso Falsos-Positivos

3. Falsos-Negativos: Caso donde se tiene a una persona, pero esta no es detectada

Figura 5.3.25: Ejemplo de un caso Falsos-Negativos

Pruebas con Haar Cascade

En la figura 5.3.26 se muestra una gráfica comparando los casos Verdaderos-Positivos, Falsos-Positivos y Falsos-Negativos, sobre las 8 muestras mencionadas anteriormente, implementando Haar Cascade.

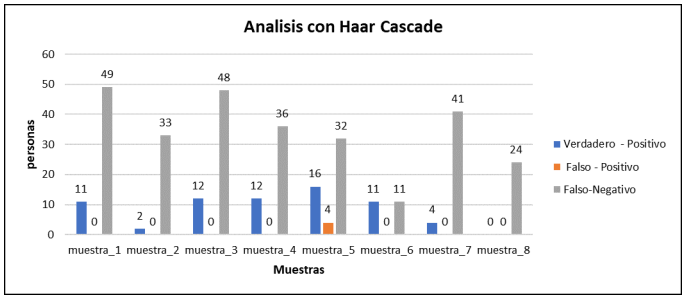


Figura 5.3.26: Analisis de Verdaderos-Positivos, Falsos-Positivos y Falsos-Negativos con Haar Cascade

Pruebas con HOG

Para la figura 5.3.27 tenemos una gráfica comparando los casos Verdaderos-Positivos, Falsos-Positivos y Falsos-Negativos, nuevamente sobre las 8 muestras mencionadas anteriormente, implementando HOG.

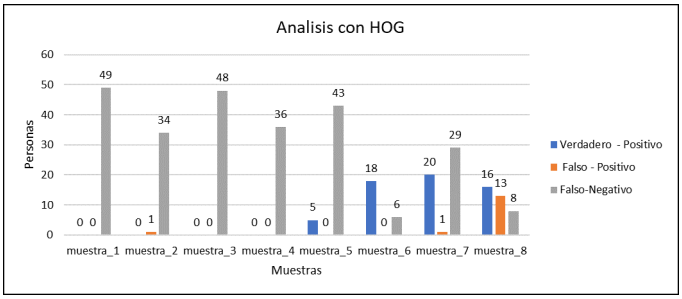


Figura 5.3.27: Analisis de Verdaderos-Positivos, Falsos-Positivos y Falsos-Negativos con HOG

5.3.5. Algoritmo con Aprendizaje profundo

Luego de esto es necesario conocer los resultados que tendría un algoritmo más complejo, en este caso se implementará un algoritmo basado en aprendizaje profundo.

Uno de los principales inconvenientes que tienen los algoritmos basados en aprendizaje profundo es la de que requieren de un alto poder de procesamiento para funcionar de manera óptima, esto se verá reflejado en las gráficas siguientes, nuevamente se mantuvo un enfoque de pocos FPS y una resolución baja.

El algoritmo que utilizaremos será el de Faster R-CNN pre entrenado con el conjunto de datos Common Objects in Context (A esto simplemente le llamaremos COCO).

Análisis en función del tiempo

En la figura 5.3.28 se muestra una gráfica del tiempo de procesamiento en función de 3 algoritmos, Haar cascade, Hog y Faster R-CNN (COCO) en el conjunto de datos BEHAVE, mientras que en la figura 5.3.29 se muestra otra gráfica del tiempo de procesamiento en función de 3 algoritmos, Haar cascade, Hog y MTCNN en el conjunto de datos CamNet.

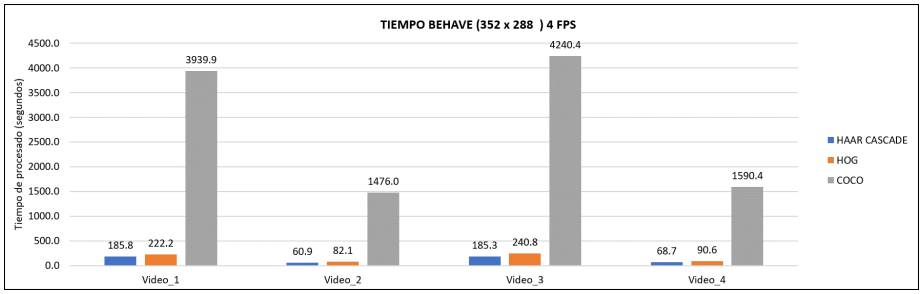


Figura 5.3.28: Gráfica de comparación entre algoritmos de aprendizaje profundo y aprendizaje de maquina convencional en función del tiempo (BEHAVE)

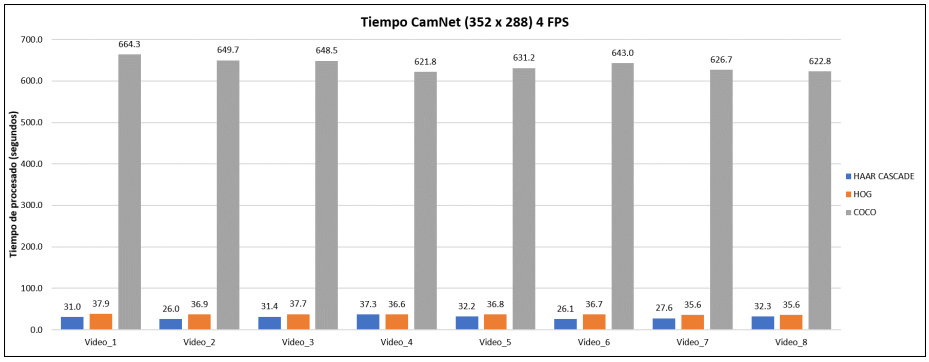


Figura 5.3.29: Gráfica de comparación entre algoritmos de aprendizaje profundo y aprendizaje de maquina convencional en función del tiempo (CamNet)

De esto se concluye que el tiempo de procesamiento en el caso de los algoritmos basados en aprendizaje profundo son sumamente altos comparados a los de aprendizaje de maquina convencional al punto de ser considerados prácticamente inviables para su utilización en tareas de videovigilancia con equipos de cómputo no muy poderosos.

Aunque una de las principales ventajas es la alta precisión que estos tienen, respecto a los primeros algoritmos estudiados, veremos cómo se comporta en términos de precisión el algoritmo basado en aprendizaje profundo.

Análisis en función de las personas detectadas

En la figura 5.3.30 se muestra una gráfica de las personas detectadas en función de 3 algoritmos, Haar cascade, Hog y Faster R-CNN (COCO) en el conjunto de datos BEHAVE, mientras que en la figura 5.3.31 se muestra otra gráfica de las personas detectadas en función de 3 algoritmos, Haar cascade, Hog y MTCNN en el conjunto de datos CamNet. La barra azul se corresponde con las personas totales identificadas en cada video.

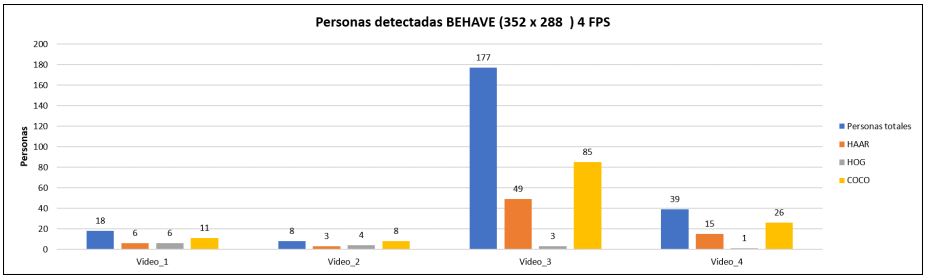


Figura 5.3.30: Gráfica de comparación entre algoritmos de aprendizaje profundo y aprendizaje de maquina convencional en función de las personas detectadas (BEHAVE)

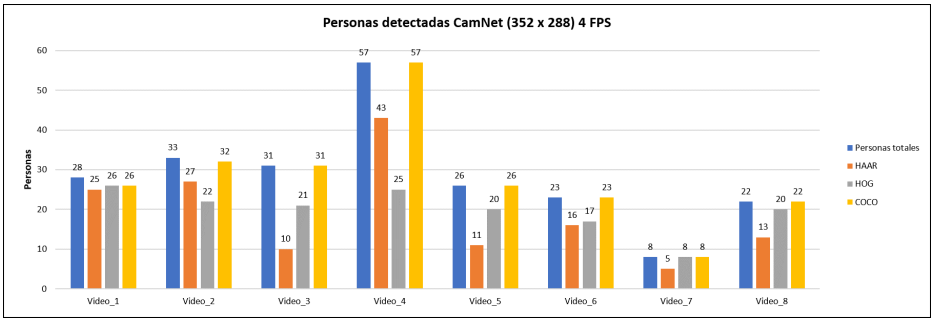


Figura 5.3.31: Gráfica de comparación entre algoritmos de aprendizaje profundo y aprendizaje de maquina convencional en función de las personas detectadas (CamNet)

Una observación pertinente de mencionar es el hecho de que este algoritmo en específico. tiene problemas al detectar personas en áreas extensas, es decir, espacios exteriores.

Casos Verdaderos-Positivos, Falsos-Positivos, Falsos-Negativos

Ahora pasemos a analizar los casos Verdaderos-Positivos, Falsos-Positivos y Falsos-Negativos con el algoritmo de aprendizaje profundo de la misma manera que se realizó para los algoritmos de aprendizaje de maquina convencional, en la figura 5.3.32 podemos ver una gráfica donde se muestran los casos anteriormente mencionados.

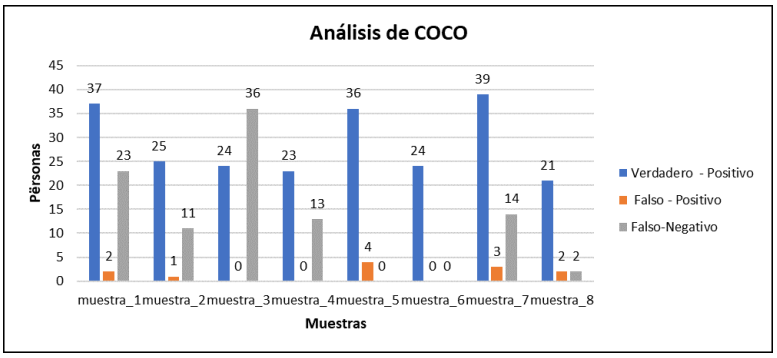


Figura 5.3.32: Análisis de casos Verdaderos-Positivos, Falsos-Positivos, Falsos-Negativos

5.4. Experimentación con detección de rostros

Con lo visto en la detección de personas, se tomarán los mismos parámetros con los que se realizaron los experimentos anteriores. Pues como se espera una mayor precisión con la detección de personas que de rostros, esta última será dependiente de la primera, por lo que nos quedaremos con los siguientes parámetros:

- Resolución a 252 x 288 pixeles
- 4 cuadros por segundo

Por lo que se medirá el tiempo de procesamiento y precisión de los algoritmos utilizando los parámetros anteriores esta vez de una forma diferente, pues al tener resoluciones fijas, así como cuadros por segundos fijos, la mejor forma de comparar tanto el tiempo requerido como la precisión, será por medio de la agrupación por conjuntos de datos como se muestra a continuación. También se medirá en conjunto el tiempo de procesamiento de los algoritmos basados en técnicas de aprendizaje de maquina convencional

- Haar Cascade
- HOG

Y la basada en aprendizaje profundo

- MTCNN

5.4.1. Tiempo de procesamiento

En las figuras 5.4.1, 5.4.2 y 5.4.3 se muestran graficas comparando el tiempo de procesamiento de 3 algoritmos, Haar cascade, Hog y MTCNN, dichas gráficas se encuentran agrupadas por conjunto de datos.

Ahora pasaremos a medir la precisión según el número de personas detectadas por cada algoritmo. Agrupadas nuevamente por conjunto de datos.

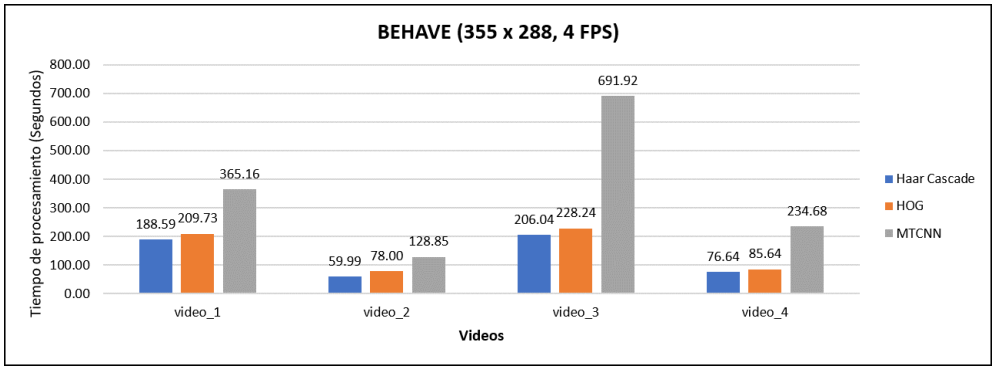


Figura 5.4.1: Gráfica de tiempo de procesamiento en la detección de rostro (BEHAVE)

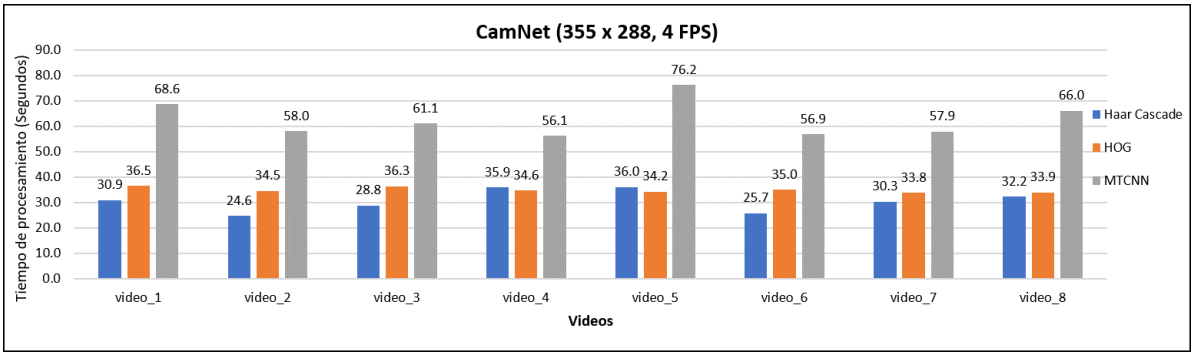


Figura 5.4.2: Gráfica de tiempo de procesamiento en la detección de rostro (CamNet)

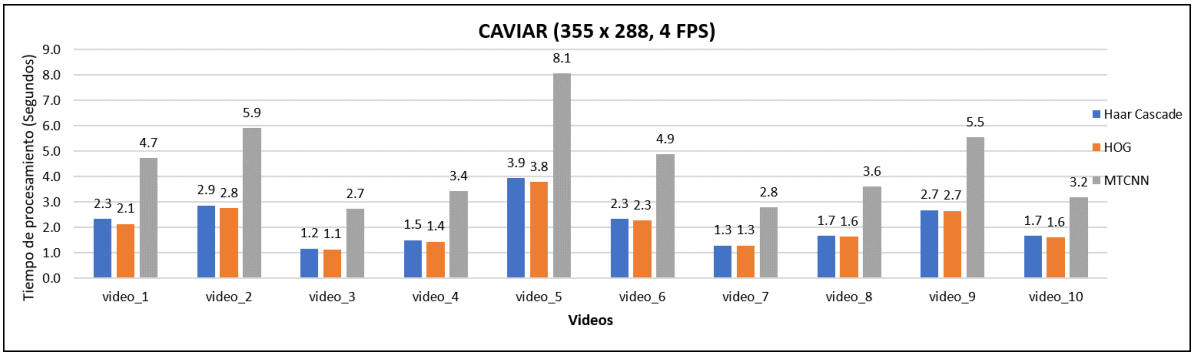


Figura 5.4.3: Gráfica de tiempo de procesamiento en la detección de rostro (CAVIAR)

5.4.2. Medición en la detección de rostros de los algoritmos

En las siguientes gráficas correspondientes las figuras 5.4.4, 5.4.5 y 5.4.6 se presenta una comparativa en la cantidad de rostros que los algoritmos llegaron a detectar por conjunto de datos, las especificaciones en cuanto a la resolución del video y la cantidad de cuadros por segundo se especifica en el título de estas. En este caso, se omite la barra de "Personas totales" o "Rostros totales", pues haciendo un análisis en los videos pues al tratarse del reconocimiento de rostros la puesta en pantalla de estos es más complicado. Así que, para simplificar las cosas, en estás gráficas, entre más alto sea el resultado, mejor.

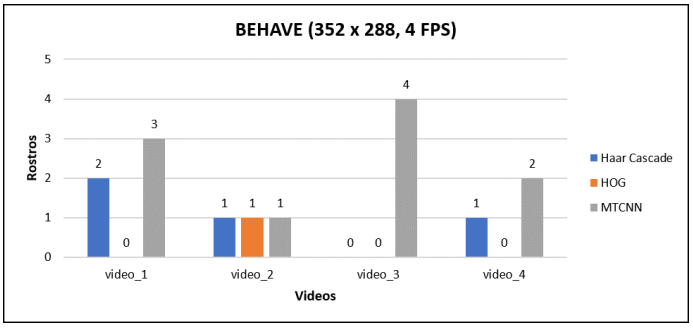


Figura 5.4.4: Gráfica de la detección de rostros por conjunto de datos(BEHAVE)

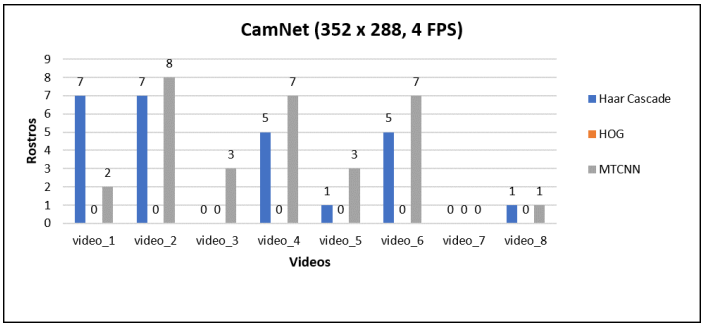


Figura 5.4.5: Gráfica de la detección de rostros por conjunto de datosCamNet)

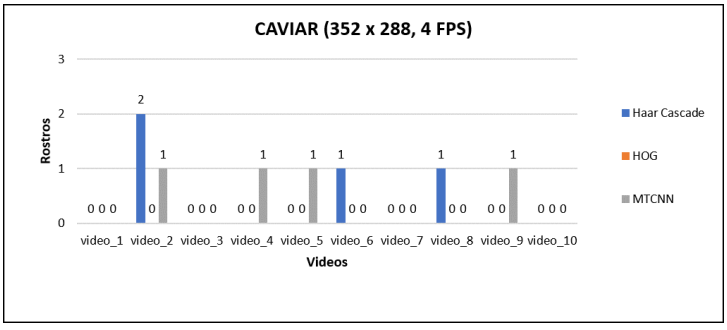


Figura 5.4.6: Gráfica de la detección de rostros por conjunto de datos(CAVIAR)

Es importante notar el pobre desempeño que obtuvo el algoritmo HOG para la detección de rostros. Eso explicaría también el porqué de su bajo tiempo de procesamiento, pues como se dejó ver en el análisis del tiempo de procesamiento para algoritmos de detección de rostros, HOG tenía un tiempo de procesamiento menor al de Haar cascade y es porque básicamente no se estaba activando.

Lo siguiente de interés sería saber qué tipos de algoritmos requieren mayor tiempo de procesamiento, para esto tomaremos los resultados arrojados por la detección de personas, esta vez agrupados por conjunto de datos, y los compararemos con los resultados arrojados por la detección de rostros.

5.5. Detección de rostros vs detección de siluetas

Lo siguiente es comparar el tiempo de procesamiento en función de la tarea de detección que estén cumpliendo detección de rostros o detección de siluetas.

Para estos experimentos, se usará la configuración de 4 fotogramas por segundo con una resolución de 355 x 288 pixeles.

En esta parte, se estarán utilizando tanto Haar Cascade como HOG para la detección de rostros y siluetas (Es decir Haar Cascade en la detección de rostros y siluetas, tanto como HOG para rostros y siluetas) en la leyenda de las gráficas se estará usando DR para detección de rostros y DS para la detección de siluetas. A su vez en el titulo de cada gráfica aparecerá de que algoritmo se trata.

5.5.1. Pruebas con BEHAVE

Correspondiéndose con las figuras 5.5.1 y 5.5.2 se puede ver la comparación del tiempo de procesamiento en función de la tarea que se esté realizando (detección de siluetas o detección de rostros), cada gráfica está agrupada por el algoritmo que se está implementando.

En la gráfica, representada por la figura 5.5.3 directamente está comparando los algoritmos MTCNN y Farster R-CNN (COCO) donde el primero se encarga de la detección de rostros, mientras que el segundo de la detección de siluetas.

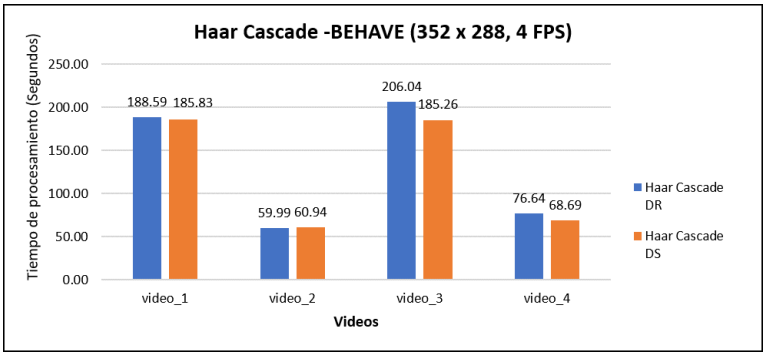


Figura 5.5.1: Gráfica de comparación entre la detección de rostros y la detección de siluetas usando Haar-Cascade

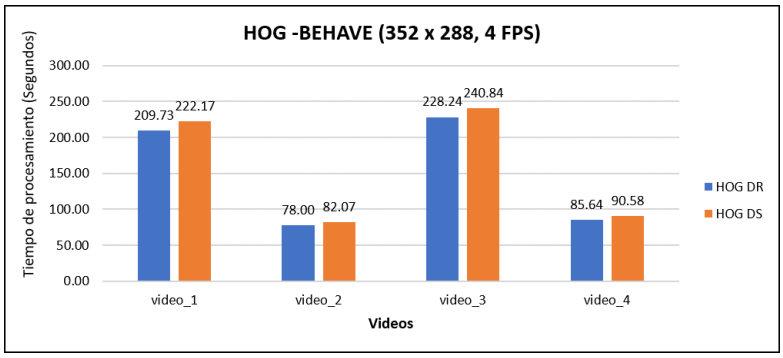


Figura 5.5.2: Gráfica de comparación entre la detección de rostros y la detección de siluetas usando HOG

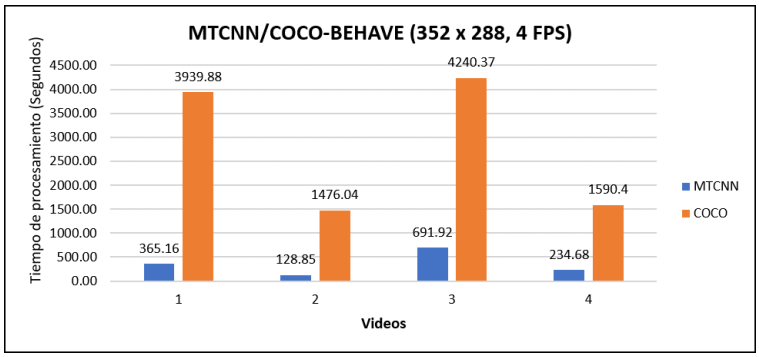


Figura 5.5.3: Gráfica de comparación entre la detección de rostros y la detección de siluetas usando MTCNN/COCO

5.5.2. Pruebas con CamNet

En las figuras 5.5.4 y 5.5.5 se puede ver la comparación del tiempo de procesamiento en función de la tarea que se esté realizando (detección de siluetas o detección de rostros), cada gráfica está agrupada por el algoritmo que se está implementando.

En la gráfica, representada por la figura 5.5.6 directamente está comparando los algoritmos MTCNN y Farster R-CNN (COCO) donde el primero se encarga de la detección de rostros, mientras que el segundo de la detección de siluetas.

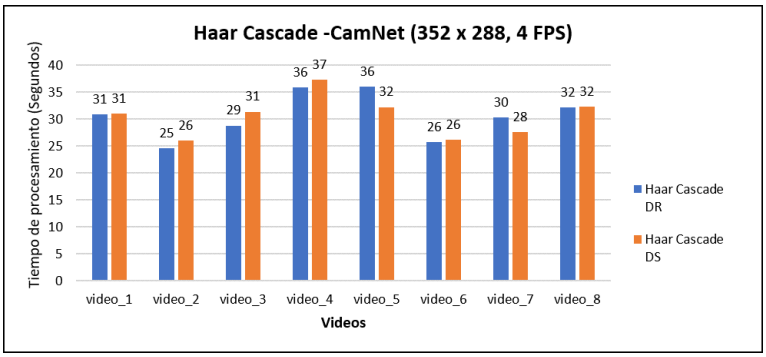


Figura 5.5.4: Gráfica de comparación entre la detección de rostros y la detección de siluetas usando Haar-Cascade

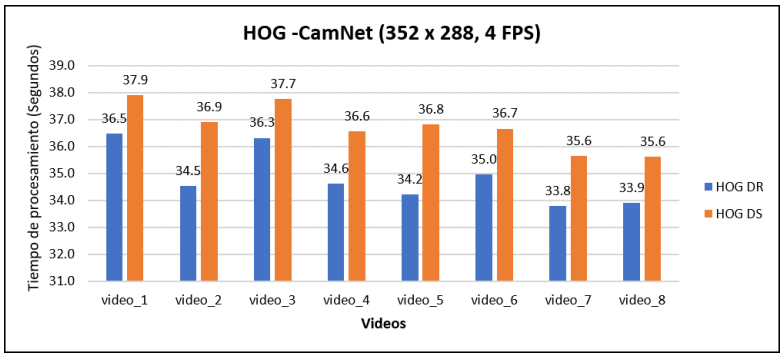


Figura 5.5.5: Gráfica de comparación entre la detección de rostros y la detección de siluetas usando HOG

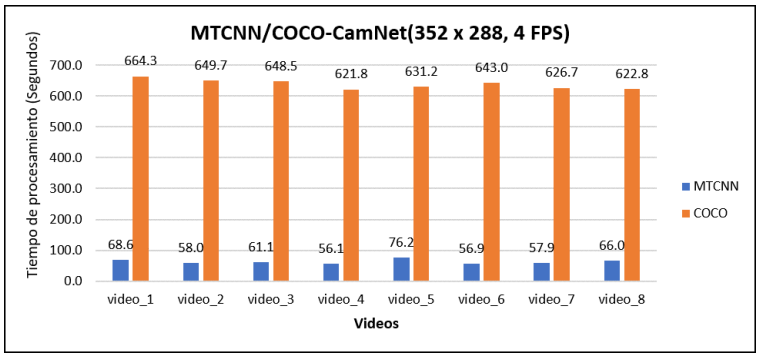


Figura 5.5.6: Gráfica de comparación entre la detección de rostros y la detección de siluetas usando MTCNN/COCO

5.5.3. Pruebas con CAVIAR

Por último, para las figuras 5.5.7 y 5.5.8 nuevamente, tenemos la comparación del tiempo de procesamiento en función de la tarea que se esté realizando (detección de siluetas o detección de rostros), cada gráfica está agrupada por el algoritmo que se está implementando. En la gráfica, representada por la figura 5.5.9 directamente está comparando los algoritmos MTCNN y Farster R-CNN (COCO) donde el primero se encarga de la detección de rostros, mientras que el segundo de la detección de siluetas.

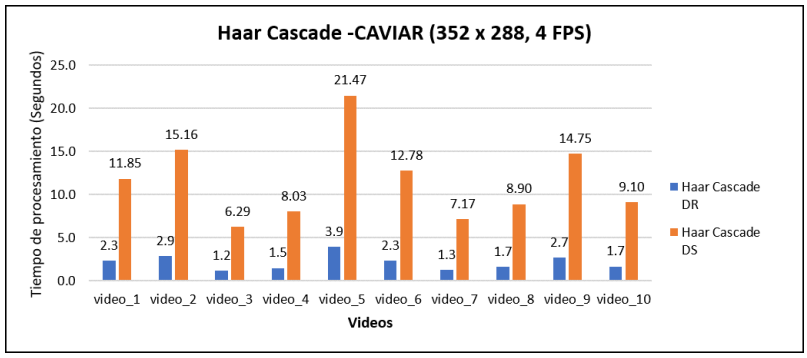


Figura 5.5.7: Gráfica de comparación entre la detección de rostros y la detección de siluetas usando Haar-Cascade

Lo anterior nos muestra claramente que los algoritmos que requieren un mayor tiempo de procesamiento son lo de detección de siluetas, esto confirma lo anteriormente mencionado al inicio de este capítulo y también concuerda

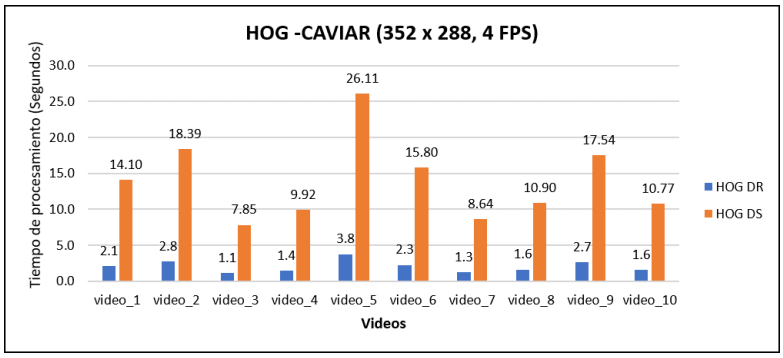


Figura 5.5.8: Gráfica de comparación entre la detección de rostros y la detección de siluetas usando HOG

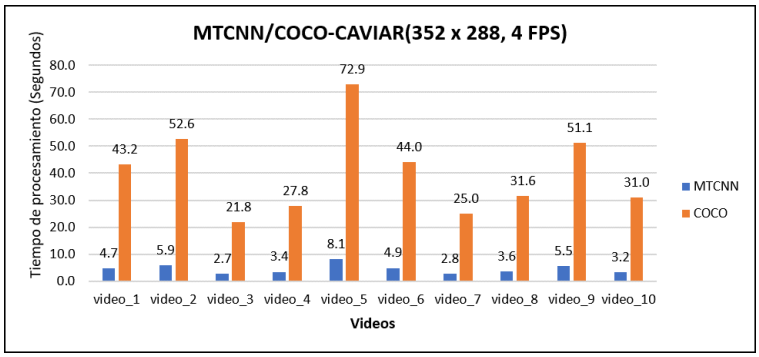


Figura 5.5.9: Gráfica de comparación entre la detección de rostros y la detección de siluetas usando MTCNN/COCO

con el marco teórico pues en la mayoría de los casos, este tiempo más alto no es por la mayor complejidad de los algoritmos de detección de siluetas, sobre los de detección de rostros, si no más bien a que estos ultimos se activan en menor medida. Esto nos da bases para que el algoritmo final, sea donde la detección de rostros dependa de la detección de siluetas.

Para esto se plantea utilizar el algoritmo de detección de siluetas de Haar Cascade y para la detección de rostros, el algoritmo de MTCNN. El cual se activará únicamente cuando Haar Cascade sea activado.

Capítulo 6

Discusión de resultados

Para presentar una solución al problema, se pretende utilizar un algoritmo de aprendizaje de maquina convencional y uno de aprendizaje profundo concretamente el algoritmo de detección de siluetas será Haar Cascade y para la detección de rostros, el algoritmo seleccionado será MTCNN. El cual se activará únicamente cuando Haar Cascade sea activado.

Con esto se pretende dotar al algoritmo de un alto porcentaje de precisión al momento de detectar un rostro, pues antes de esto quiere decir que se encontró con una silueta humana.

Para medir la efectividad de esto, se presentará un análisis en el tiempo de procesamiento de los dos algoritmos, así como de su precisión al momento de reconocer un rostro donde prácticamente se pretende asegurar la existencia de un ser humano.

6.1. Análisis general del tiempo de procesamiento

Para tener una idea de cuan efectivo es el algoritmo en tiempo de procesamiento veremos cómo se comporta comparado con el algoritmo de Haar Cascade y el de MTCNN, es lógico pensar que, de menos, se demuestre que este tenga un tiempo superior al de Haar Cascade, para apoyar lo dicho, se tienen las figuras 6.1.1, 6.1.2 y 6.1.3, que corresponden a gráficas donde se compara el tiempo de procesamiento por algoritmo.

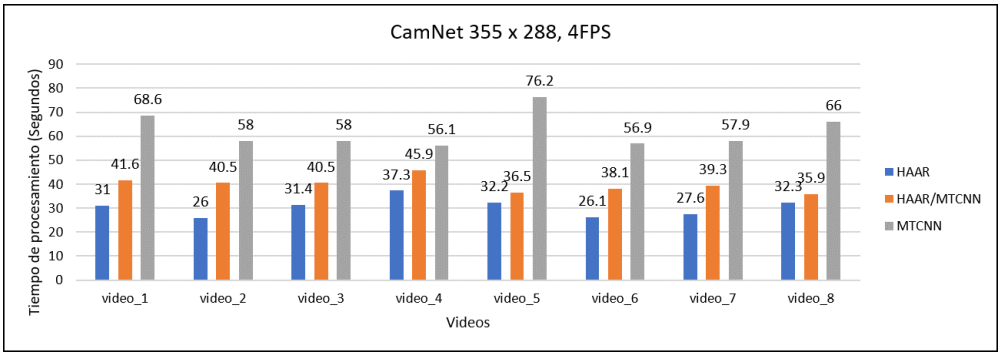


Figura 6.1.1: Gráfica de comparación entre Haar Cascade vs Haar cascade/MTCNN vs MTCNN (CamNet)

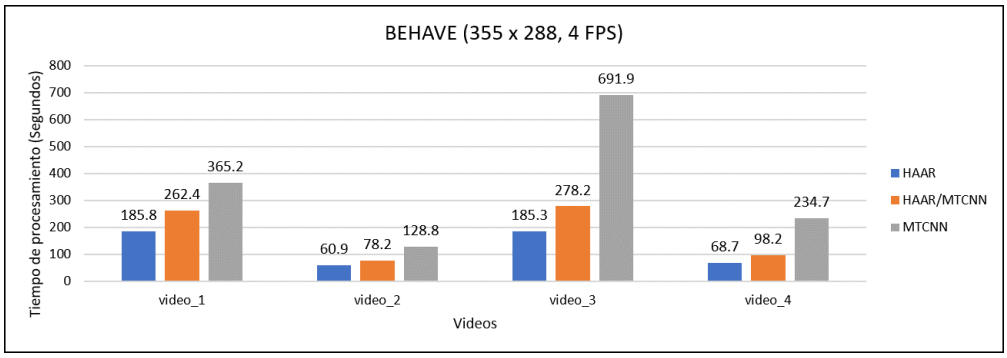


Figura 6.1.2: Gráfica de comparación entre Haar Cascade vs Haar cascade/MTCNN vs MTCNN (BEHAVE)

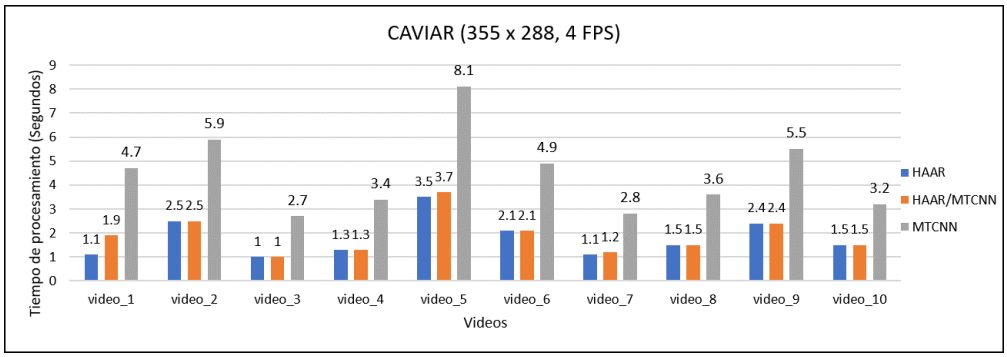


Figura 6.1.3: Gráfica de comparación entre Haar Cascade vs Haar cascade/MTCNN vs MTCNN (Caviar)

Esto se puede ver de forma más directa en la figura 6.1.4, donde directamente se compara el tiempo de procesamiento con todos los conjuntos de datos.

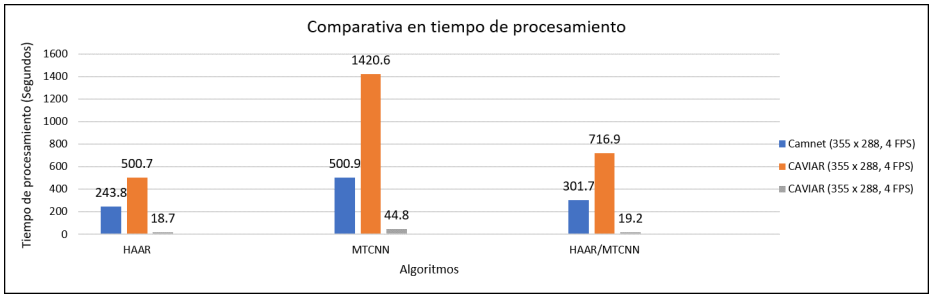


Figura 6.1.4: Gráfica de comparación entre Haar Cascade vs Haar cascade/MTCNN vs MTCNN

En la figura 6.1.5 se nos muestra como HAAR/MTCNN es casi el doble de rápido que MTCNN, pues el primero, tomando en cuanta los tres conjuntos de datos, obtuvo un tiempo de procesamiento de 1037 segundos, mientras que MTCNN, obtuvo uno de 1966 segundos.

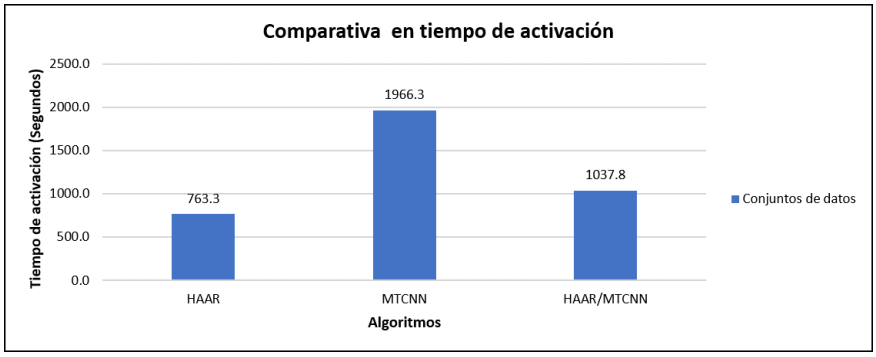


Figura 6.1.5: Gráfica de Tiempo de activación total por algoritmo

Podemos ver que seguir esta metodología de activar el algoritmo de deep learning, solo cuando se haya activado el algoritmo de aprendizaje de maquina convencional, presenta ventajas sobre dejar correr ambos algoritmos por separado donde, por ejemplo, MTCNN este detectando todo el tiempo. También como se obtiene una gran ganancia en ambientes que cubren largas distancias, como podría ser en los dos últimos casos del conjunto de datos BEHAVE.

Una forma en la que podemos notar si el algoritmo es una buena opción para el procesamiento de imágenes en tiempo real sería la de comprobar cuanto tiempo tarda en procesar todos los conjuntos de datos en comparación de la duración, de los mismos. Para ello se presenta la gráfica 6.1.6.

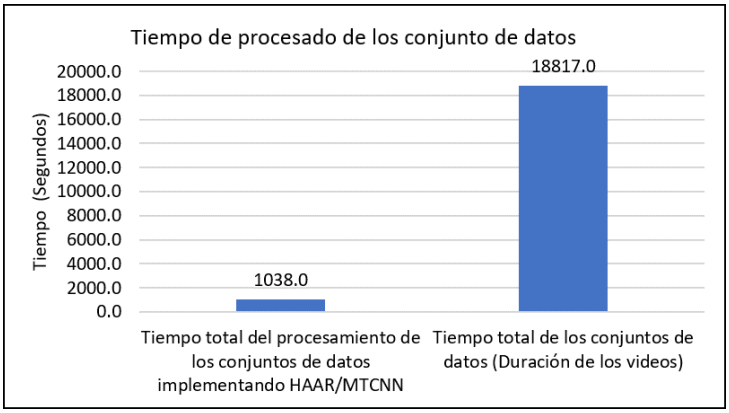


Figura 6.1.6: Comparación entre el tiempo de procesamiento de HAAR/MTCNN con la duración de los conjuntos de datos

Con esos datos podemos hacer las siguientes estimaciones, si el tiempo total de los videos contenidos en nuestro conjunto de datos, corresponden a 10820 segundos estado a 4 cuadros por segundo eso nos da un total de 75268 cuadros que contiene todo nuestro conjunto de datos y nuestro tiempo de procesamiento total implementando el algoritmo HAAR /MTCNN fue de 1038 segundos, lo que quiere decir que nuestro algoritmo fue capaz de procesar esos 75268 cuadros en 1038 segundos, siendo así, nuestro algoritmo debió haber procesado 72 cuadros en un segundo, donde dentro de ese lapso de procesamiento, se podrían (o no) haber activado los algoritmos de detección.

Aunque el número parece impresiónate, se debe tener en cuenta que esta estimación se basa en conjuntos de datos a una resolución muy pequeña y bajos cuadros por segundo. Si pensamos en entornos más realistas. las cámaras

de seguridad comúnmente gravan a más de 4 cuadros por segundo, teniendo algunas que lo hacen a 24, 30 e incluso 60 cuadros por segundo, y si se quisiera considerar un muestreo sistemático para "reducir" el número de FPS, al tiempo de procesamiento del algoritmo se le debe agregar el tiempo de ese proceso, así como también el de redimensionamiento, pues volviendo a mencionar las cámaras de videovigilancia, estás operan a resoluciones mayores.

Dicho lo anterior, es indudable querer conocer cuanto afecta esta aproximación al algoritmo de detección de rostros en términos de su precisión. Por lo que tendremos que analizar, en primera instancia que tan reducida se ve su activación, para poder tener una idea de cuanto afectó el hecho de tener un "filtro" o una condición extra para activarse. Por lo que procederemos a analizar las activaciones de MTCNN y de HAAR/MTCNN.

6.2. Comparación entre la activación en MTCNN y HAAR/MTCNN

En las siguientes gráficas, correspondientes a las figuras 6.2.1, 6.2.2 y 6.2.3, se verá cómo fue la activación de los algoritmos por conjunto de datos, lo que se espera ver, es que las activaciones de MTCNN bajen considerablemente por el filtro Haar cascade, estás gráficas se encuentran ordenadas por conjunto de datos.

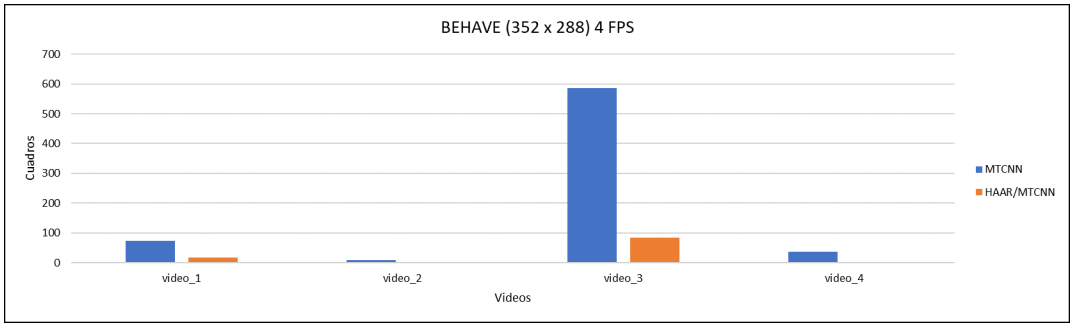


Figura 6.2.1: Gráficas de activación, Haar cascade/MTCNN vs MTCNN (BEHAVE)

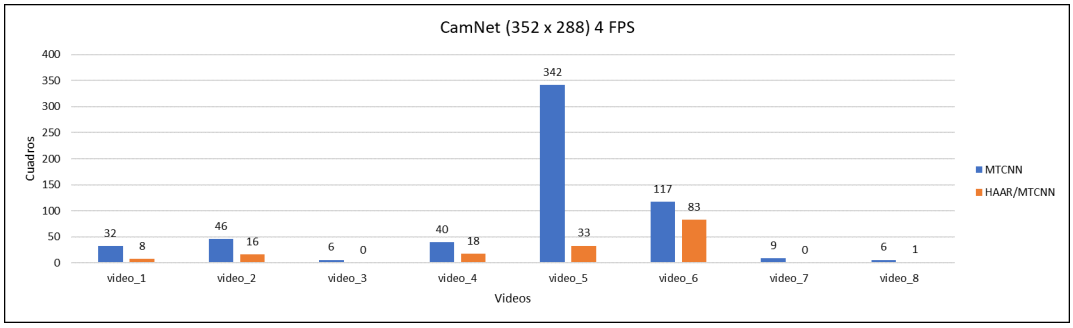


Figura 6.2.2: Gráficas de activación , Haar cascade/MTCNN vs MTCNN (CamNet)

Se puede notar al ver la figura 6.2.4 que la activación se ve drásticamente reducida, específicamente MTCNN obtuvo un total de 1328 mientras que Haar/MTCNN dio un total de 264. cuadros activados, esto sumando las activaciones de los tres casos.

Lo que se traduce en que se redujo en cinco, el total de cuadros de activación o qué el algoritmo MTCNN se activó únicamente una quinta parte de lo que se activaría sin el filtro de HAAR, pero es importante mencionar que se

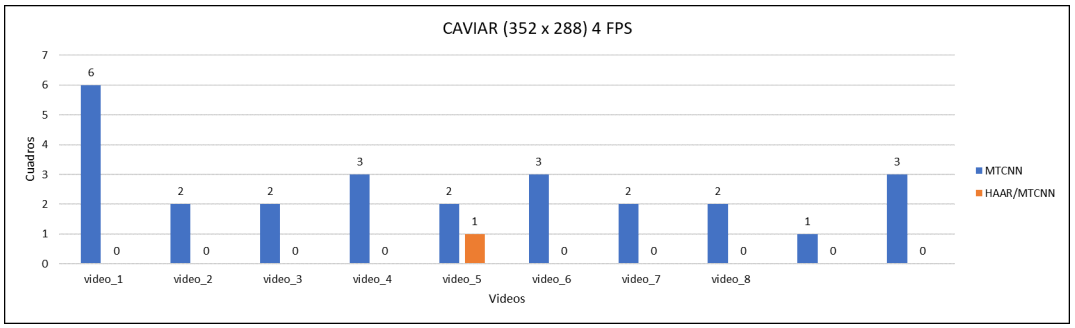


Figura 6.2.3: Gráficas de activación , Haar cascade/MTCNN vs MTCNN (CAVIAR)

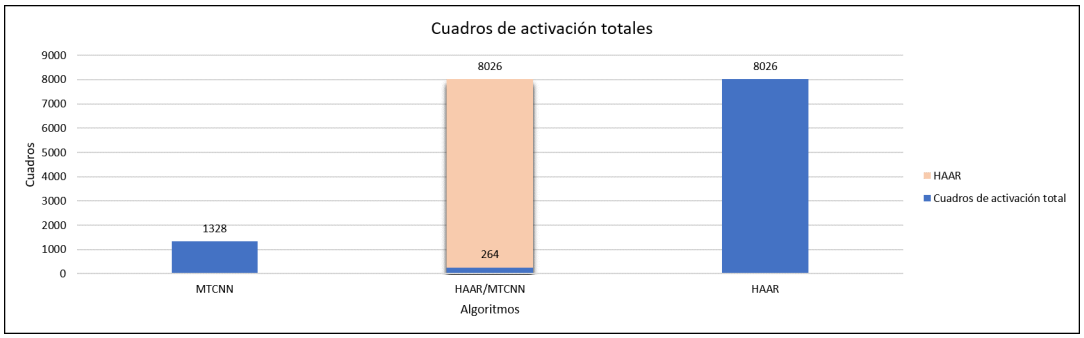


Figura 6.2.4: Gráficas de activación total, Haar cascade/MTCNN vs MTCNN

están contando únicamente las activaciones del MTCNN luego de aplicar el filtro, pues las activaciones del Haar Cascade, serían las mismas.

Capítulo 7

Conclusiones

A lo largo de este trabajo de investigación se ha buscado por una herramienta que haga frente a los problemas inseguridad que aquejan a la población día con día, esto mediante sistemas de videovigilancia. Por ello se recurrió a la visión por computadora la cual puede ser fácilmente aplicable a cámaras de videovigilancia con el fin de realizar un trabajo más preciso y sobre todo convirtiendo a las cámaras en un medio activo de seguridad. Esto despierta la necesidad de aplicar algoritmos capaces de detectar siluetas y rostros, un área en el que la inteligencia artificial lleva aplicándose durante ya un buen tiempo. Específicamente, en esta investigación se optó por el uso de aprendizaje de máquina convencional, así como su variante más "moderna", conocida como aprendizaje profundo. Siendo así que sea vital encontrar algoritmos que sean capaces de realizar tareas detección de siluetas y rostros tomando siempre en cuenta el costo de implementación computacional. Es precisamente el coste computacional, el que nos limita al momento de implementar algoritmos complejos y por ende más precisos. Con estas consideraciones es de suma importancia encontrar un balance entre la precisión de un algoritmo y su coste computacional, pues si se desea en algún momento realizar una implementación de la solución o expandir la misma, esta debe ser capaz tener un buen desempeño, aún si no se trata de un sistema de cómputo poderoso.

Otro punto de suma importancia que se trabajó en esta investigación, es la fuente de video en sí, es decir las características que debía tener la entrada de nuestro algoritmo, por las razones ya expuestas en los capítulos anteriores se hace notar que la resolución así como los cuadros por segundo de cada video, son básicamente lo que dictamina que tan eficiente es un algoritmo, es por eso que se realizaron una gran cantidad de pruebas, para determinar un mínimo de resolución así como de cuadros por segundo además se realizó un cálculo de la velocidad promedio de una persona al caminar para estimar un mínimo de cuadros por segundo.

De esta manera la configuración encontrada de 5 cuadros por segundo a una resolución de 352 x 288 pixeles fue la que resulto con un buen balance entre la precisión y el tiempo de procesamiento a pesar de ser considerablemente menores a las resoluciones y cuadros por segundo que usualmente estamos acostumbrados de ver.

Aunque aún con una disminución sumamente considerable en la calidad de la entrada de video, se hizo evidente que algunas tecnologías que hacen uso de aprendizaje profundo son aún muy complejas para los procesadores de uso doméstico (véase la serie I de Intel o la serie Ryzen de AMD) , la muestra más clara de esto es la implementación de Faster-R CNN, el cual presenta grandes tiempos de procesamiento al momento de implementarse en procesadores, aunque sin duda, su precisión fue la más alta en todos los escenarios y se recomienda su implementación apoyada de una unidad gráfica de procesamiento.

Esto nos dice que su implementación de manera eficiente se queda lejos de las máquinas que no cuentan con

hardware dedicado al procesamiento gráfico. Los resultados arrojados con el algoritmo propuesto Haar cascade para detección de siluetas y MTCNN para la detección de rostros, ejemplifican como se puede apoyar una tecnología más compleja y robusta, en una más simple, para de cierta forma, hacer un balance entre tiempo de procesamiento y precisión de los algoritmos. Pues como se pudo apreciar en el capítulo anterior se da sustento para la aplicación de Haar cascade como el algoritmo que determina la activación de MTCNN. Con esto se logra una disminución en el tiempo de procesamiento, pues MTCNN no se está activando durante todo el procesado del vídeo, y cuando lo hace, es más probable que encuentre un rostro. Aunque claro, ahora se tiene el inconveniente de que forzosamente se necesita detectar una silueta antes que un rostro, provocando un incremento de falsos-negativos y que los falsos-positivos, conlleven a un mayor coste computacional pues prácticamente estarían activando dos algoritmos, donde el segundo, no encontrará nada.

Aún con esos inconvenientes, los algoritmos demostraron tener una precisión apreciable esto gracias también, a los escenarios donde estos fueron estudiados. Una gran ventaja de los escenarios utilizados son que predominaba la buena iluminación, áreas extensas e incluso áreas que hacían favorable la detección de rostros. Por ello queda como trabajo futuro el estudio de los algoritmos en áreas con menos de estas ventajas, que a su vez asemejen a escenarios similares a los de un delito, como una calle o habitación oscura. Teniendo en cuenta que año con año las unidades de procesamiento se vuelven más poderosas y de igual manera, los algoritmos de reconocimiento de patrones se hacen más robustos, nos da la posibilidad de encontrar nuevas formas de resolver tareas de videovigilancia, hasta un punto donde se vuelva potencialmente comercial aunque, como se demostró en esta investigación, no se tiene porque utilizar los algoritmos más recientes para dar buenas soluciones, si no, el de investigar cuales son los más eficientes tomando a consideración el balance entre el costo, tanto computacional y monetario al hablar de la inversión en un equipo de cómputo, con el beneficio que se espera. De manera general podemos concluir que la implementación de sistemas de videovigilancia haciendo uso de inteligencia artificial presentan grandes ventajas sobre sistemas convencionales, aunque presentan el inconveniente de requerir unidades de procesamiento y este problema se acentúa cuando se opta por implementar algoritmos de aprendizaje profundo. Es por esto, que es de vital importancia realizar investigaciones paulatinas en esta área para monitorear el avance pertinente a las capacidades de las unidades de procesamiento junto con la optimización de los algoritmos, de esta forma podrían llegar a darse con soluciones ideales para la masificación y comercialización a un costo más asequible.

Como trabajo futuro sería una buena opción agregar más complejidad al algoritmo y de esta manera que realice tareas más complejas, como la detección de armas de fuego, estimar altura de las personas identificadas o incluso detectar si se trata de un hombre o una mujer o incluso agregar nuevos algoritmos, como de iluminación artificial de áreas oscuras. También se pueden presentar ideas de aplicación fuera del ámbito de la seguridad, pudiendo aplicarse también a otras áreas, como la domótica. Aunque también se podrían tomar únicamente las métricas de la entrada para comparar otros algoritmos candidatos a la solución de identificación se siluetas y/o rostros.

Bibliografía

[1] R. D. Adrian. Deep learning for computer vision with python, 2017. 10, 14, 18

[2] A. H. Angmin Oh. Virat video dataset. Accedido el 28-11-2020 a url https://viratdata.org/, 2011. 36

[3] F. Azorín and J. L. Sánchez-Crespo. *Métodos y aplicaciones del muestreo*. Alianza Madrid, 1994. 34

[4] M. A. Boden and J. C. A. Sanjose. *Inteligencia artificial y hombre natural*. Tecnos Madrid, 1984. 13

[5] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585, 2015. 28

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005. 26

[7] A. Dantcheva, C. Velardo, A. D'angelo, and J.-L. Dugelay. Bag of soft biometrics for person identification. *Multimedia Tools and Applications*, 51(2):739–777, 2011. 23

[8] L. E. C. Díaz. Consideraciones en torno a las posibilidades de implantación de los sistemas expertos en las decisiones organizativas. *Esic market*, (65):77–90, 1989. 13

[9] S. S. Farfade, M. J. Saberian, and L.-J. Li. Multi-view face detection using deep convolutional neural networks. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 643–650, 2015. 29

[10] R. Fisher. Caviar test case scenarios. Accedido el 28-11-2020 a url https://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/, 2007. 36

[11] S. Z. G. S. Chrysos, E. Antonakos and P. Snape. 300 videos in the wild (300-vw) challenge 'i&' workshop (iccv 2015). Accedido el 28-11-2020 a url https://ibug.doc.ic.ac.uk/resources/300-VW/, 2015. 36

[12] S. Z. G. S. Chrysos, E. Antonakos and P. Snape. Human pose and path estimation from aerial video using dynamic classifier selection," cognitive computation. Accedido el 28-11-2020 a url https://asankagp.github.io/aerialgaitdataset/, 2015. 36

[13] F. general del Estado de Puebla. Incidencia delictiva por municipio. Accedido en 11-03-2021 a urlhttps://fiscalia.puebla.gob.mx/index.php/informacion-socialmente-util/incidencia-delictiva-por-municipio/, 2019. 1

[14] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 27

[15] A. J. Goldstein, L. D. Harmon, and A. B. Lesk. Identification of human faces. *Proceedings of the IEEE*, 59 (5):748–760, 1971. 23

[16] G. INEGI (Instituto Nacional de Estadistica. Encuesta nacional de victimización y percepción sobre seguridad pública (envipe), 2020.

[17] A. Laghaee. Behave interactions test case scenarios. Accedido el 28-11-2020 a url http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/INTERACTIONS/, 2007. 36

[18] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997. 23

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 29

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 27

[21] D. Matthew and R. Fergus. Visualizing and understanding convolutional neural networks. In *Proceedings of the 13th European Conference Computer Vision and Pattern Recognition, Zurich, Switzerland*, pages 6–12, 2014. 28

[22] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943. 14, 22

[23] J. T. P. Méndez and R. M. Morales. *Inteligencia artificial*. 2008. 13

[24] V. S. Nalwa. *A guided tour of computer vision*. Addison-Wesley Longman Publishing Co., Inc., 1994. 21

[25] C. N. Niépce. Nicéphore niépce y daguerre. Accedido en 15-01-2020 a urlhttp://photo-museum.org/es/historia-fotografia/, 2019. 22

[26] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 27

[27] L. G. Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963. 22

[28] F. Rosenblatt. Principles of neurodynamics: Perceptions and the theory of brain mechanisms. 1962. 22

[29] T. F. S. Zhang, E. Staudt and A. Roy-Chowdhury. A camera network tracking (camnet) dataset and performance baseline. Accedido el 28-11-2020 a url https://drive.google.com/drive/folders/0B7uDdIqGrZlVfmdlOFZyUEg3RHUxaUdSaGVGTTNDT3R0dUNLSFdCSkZYVWk0dE16TFg4cTA, 2015. 36

[30] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 23

[31] R. Sethi. Caviar test case scenarios. Accedido el 28-11-2020 a url https://data.mendeley.com/datasets/sn9yxn5zsz/1, 2007. 36

[32] G. SIMONS. Introducing artificial intelligence (introducción a la inteligencia artificial, traducción de andrés magaña. 1984. 13

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 28

[34] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Josa a*, 4(3): 519–524, 1987. 23

[35] M. Turk, A. Pentland, P. Belhumeur, and J. Hespanha. Eigenfaces for recognition: Journal of cognitive neurosicence. 1991. 23

[36] L. Uhr. "pattern recognitionçomputers as models for form perception. *Psychological Bulletin*, 60(1):40, 1963. 23

[37] V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013. 27

[38] P. Viola and M. Jones. Robust real-time face detection. In *null*, page 747. IEEE, 2001. 23

[39] D. Weinland, R. Ronfard, and E. Boyer. Ixmas actions – new views and occlusions. Accedido el 28-11-2020 a url https://www.epfl.ch/labs/cvlab/data/data-ixmas10/, 2007. 36

[40] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 29

[41] S. Zhou, V. Krueger, and R. Chellappa. Probabilistic recognition of human faces from video. *Computer vision and image understanding*, 91(1-2):214–245, 2003. 23