UNIVERSIDAD AUTÓNOMA DE ZACATECAS

"Francisco García Salinas"

"Detección de Infantes en la Zona de copiloto en un Vehículo, mediante Redes Neuronales Profundas y Visión Computacional"

Tesis para obtener el grado de:

Maestro en Ciencias del Procesamiento de la información

Presenta

William Castañeda Almaraz

Director

Dr. José María Celaya Padilla

Co-Director

Dr. Huizilopoztli Luna García

Zacatecas, Zac., Diciembre de 2020

Resumen

Los accidentes automovilísticos generan millones de muertes en el mundo, y lesiones a infantes de acuerdo con la Organización Mundial de la Salud (OMS), la detección de un infante como copiloto en un vehículo podría prevenir lesiones o fallecimientos de infantes en accidentes automovilísticos. Por ello, la presente investigación describe el uso de redes neuronales convolucionales para la detección de infantes en el asiento del copiloto en automóviles mediante la implementación de un sistema de visión computacional. El proceso que se siguió para el desarrollo de proyecto fue la adquisición de las imágenes de los casos a estudiar, el pre-procesamiento de las imágenes y en el reentrenamiento de la red se empleó la técnica de “ajuste fino” basado en la arquitectura Google Net de Redes Neuronales Convolucionales, se evaluó el rendimiento del modelo mediante un análisis estadístico multiclase obteniendo una exactitud 97.66%, sensibilidad promedio 97.44% y especificidad promedio 98.95%.

Abstract

Dedicatorias

Para comenzar dedico el presente trabajo a Dios por permitirme llegar a la culminación y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

A mis padres Norman Castañeda y Sandra Almaraz, a quienes les agradezco por su amor, trabajo y sacrificio en todos estos años, pues gracias a ustedes he logrado realizarme profesionalmente y convertirme en lo que soy.

Agradecimientos

A todas las personas que han estado a mi lado durante el transcurso de esta etapa de mi vida, les agradezco el apoyo incondicional, sus consejos y los nuevos conocimientos que me otorgaron e impulsaron para crecer como persona.

A mis padres y hermana

Por haberme apoyado en todo momento, por los consejos, las bendiciones, el amor, que han sido factores esenciales para guiarme por el buen camino, y que me ha permitido ser una persona de bien.

Directores de tesis

Por su esfuerzo y dedicación al compartir sus conocimientos. Agradezco su manera de trabajar, su persistencia, su paciencia, sus orientaciones y su motivación han sido fundamentales para mi formación.

A los docentes

Quienes formaron parte de mi desarrollo, por todas sus enseñanzas, consejos recibidos a lo largo de los últimos años que marcaron siempre un punto clave en este proyecto, gracias por compartir sus conocimientos y hacer de esta investigación lo que es hoy en día.

A mis amigos

Estaré eternamente agradecido a mis compañeros de clase Joyce Selene Anaid Lozano Aguilar, Javier Saldívar Pérez, Luis Carlos, Reveles Gómez, Martın Hazael Guerrero Flores. El ambiente de trabajo creado es el mejor, su visión, motivación y optimismo me han ayudado en momentos críticos del presente trabajo de investigación. Los considero como mis mejores amigos y agradezco cada momento su apoyo incondicional en cada situación de clase o personal y estoy orgulloso de todos ustedes.

Índice

Lista de Figuras

Lista de Tablas

Palabras Clave

ANN Redes Neuronales Artificiales

CNN Redes Neuronales Convolucionales

DL Aprendizaje Profundo

TL Transferencia de Conocimiento

PDI Procesamiento Digital de Imágenes

Capítulo 1. Introducción

En el presente capítulo se describe el contexto de la investigación; Antecedentes, definiendo la problemática, la justificación de la investigación, planteando el objetivo general y objetivos específicos.

1.1. Antecedentes

En el Diario Oficial del 7 de febrero de 1984, de la Ley General de Salud y de acuerdo con la Organización Mundial de la Salud (OMS), un accidente se define como “el hecho súbito que ocasiona daños a la salud y que se produce por la concurrencia de condiciones potencialmente prevenibles” [1].

Los accidentes de tránsito son un problema de la actualidad debido a su magnitud, pero su antecedente es remoto. Se tienen datos del año 1300, en la Ciudad del Vaticano, en donde se suscitaron una gran cantidad de accidentes durante la celebración del año santo. A partir de este suceso, las autoridades decidieron tomar medidas de prevención como pintar calles, puentes y caminos con rayas blancas [2].

El primer caso registrado de muerte por accidente de tráfico se produjo en 1869 en Irlanda cuando Mary Ward falleció tras caer de un vehículo a vapor diseñado por su primo [3]. En el año 1896 un coche, impulsado a gas, atropelló a una dama de nombre Bridget Driscoll en las calles de Londres y en ese mismo año se originó la primera imposición de multa de tránsito por exceso de velocidad [4].

Los accidentes automovilísticos según la OMS, generan 1.3 millones de muertes humanas en carreteras del mundo entero y entre 20 a 50 millones de traumatismos no mortales, siendo la principal causa de defunción de infantes y jóvenes de 5-29 años [5]. En México fallecen 24 mil personas al año y 55 cada día por accidentes de tránsito, ocupando el séptimo lugar a nivel mundial [6]. En el 2018, Zacatecas fue considerado como el estado con mayor tasa de defunciones en accidentes automovilísticos de 24.7 por cada 100 mil habitantes según el INEGI (Instituto Nacional de Estadística Geográfica)[7].

Con la popularización de los vehículos y los accidentes automovilísticos la seguridad en los automóviles fue aumentado gradualmente, la empresa Ford en 1956 fue quien comenzó a introducir el cinturón de seguridad en su gama de modelos como parte de los equipamientos opcionales [8]. Otros esfuerzos para ofrecer seguridad como los reposacabezas que, en un impacto por alcance, proporciona apoyo para la cabeza, manteniendo alineadas las vértebras cervicales. Las bolsas de aire que tienen como objetivo desacelerar a los ocupantes amortiguando el impacto ayudando a mantenerlos en el interior del vehículo, y reduciendo las lesiones entre 40% y 50%. El contar con bolsas de aire en combinación con el cinturón de seguridad, reduce en un 68% el riesgo de muerte. Otro sistema de seguridad es el ISOFIX que consiste en sistema estándar de fijación de sillitas para infantes y bebés, concebido para resultar especialmente sencillo. Fue creado por la ISO - International Organization for Standardization a principios de los años 90 después de que varios estudios concluyeron que hasta en un 90% de los casos, las sillitas se montaban incorrectamente [9].

En la actualidad la Inteligencia Artificial juega un papel clave al mejorar la calidad de vida de las personas en múltiples formas, una de ellas es la seguridad en los medios de trasporte donde se podrían reducir significativamente los accidentes de tránsito y, por ende, las muertes.

En el área de clasificación y procesamiento de información existen técnicas como lo son las redes neuronales cuyo objetivo es simular un comportamiento inteligente, tratando de imitar la forma en que trabajan las neuronas biológicas. Su principal característica son su habilidad de aprender dado un conjunto de datos mediante un proceso de entrenamiento[10].

El inicio de las redes neuronales surge con los perceptrones de Rosenblatt, el cual no es una red neuronal sino, el primer algoritmo que describe el funcionamiento de las redes neuronales, fue publicado en 1958 y sus principales usos fueron par decisiones sencillas. Cabe mencionar que otro avance impórtate fue en 1989 con el trabajo de Yann LeCun con las Redes Neuronales Convolucionales, Dichas redes están inspiradas en la corteza visual de los animales y sus principales aplicaciones en el reconocimiento de video y procesamiento de la lengua.[11]

Dada la anterior introducción procedemos a describir a detalle el planteamiento del problema que se pretende resolver en el presente trabajo de tesis.

1.2. Planteamiento del Problema

En México, uno de los principales problemas en el ámbito de la movilidad es la falta de conciencia en algunos conductores al no llevar a los infantes en un lugar correcto o no contar con el equipo adecuado para la seguridad de los menores, ha provocado que los accidentes de tráfico sigan siendo una de las principales causas de muerte entre los infantes, generado pérdidas humanas y lesiones de gravedad, años perdidos prematuramente, daños emocionales, entre otros [12]. De acuerdo con datos del Instituto Nacional de Salud Pública (INSP), México ocupa el séptimo lugar a nivel mundial y el tercero en la región de Latinoamérica en muertes por siniestros viales [13]. Según datos del Consejo Nacional para la Prevención de Accidentes (CNPA) en México, los accidentes automovilísticos son la principal causa de muerte entre infantes de 4 y 12 años [14] y señala que del 75% al 90% de los fallecimientos automovilísticos pudieron ser evitados si el infante hubiese llevado un equipo con sistemas de retención infantil (SRI) adecuado para su edad y peso [15].

Los accidentes automovilísticos que involucran a infantes se han convertido en un problema mundial, debido a lo anterior, se busca la prevención de fallecimientos y lesiones en infantes en vehículos, por ello las empresas automotrices han implementado tecnologías como Inteligencia Artificial (AI), Sistemas de Visión Computacional en conjunto con sensores que son capases de detectar infantes en un vehículo. Una de estas empresas es tesla, la cual desarrollo un dispositivo de detección que por medio de antenas e inteligencia artificial detecta infantes en los vehículos, que podría ayudar a evitar que los niños se queden al interior del auto con climas extremadamente calientes [16], La empresa cybex desarrollo “SENSORSAFE”, que es un sensor encargado de monitorear la presencia de un infante y la presencia de calor excesivo al interior de un vehículo. Su principal objetivo es prevenir el fallecimiento de infantes por golpes de calor al interior del vehículo por medio de notificaciones al dispositivo móvil [17].

Los desarrollos mencionados anteriormente presentan un enfoque hacia la seguridad de infantes para la prevención de accidentes por golpes de calor. Debido a que actualmente no se han encontrado implementaciones en la industria relacionadas con la detección de infantes en el área de copiloto para evitar el fallecimiento de infantes, la presente investigación se propone la *"Implementación de un algoritmo para la Detección de Infantes en la Zona de Copiloto en el Vehículo, a través de Implementación de Redes Neuronales Profundas"* para la prevención de fallecimientos o lesiones, en accidentes viales.

1.3. Justificación

La prevención de lesiones y pérdidas humanas en accidentes de tránsito es un problema de salud que se tiene que resolver, ya que los accidentes de tránsito son la principal causa de muerte en infantes entre 4 a 12 años, además de ser una de las principales causas de discapacidad en los infantes, ya que cada año mueren 260,000 infantes y sufren lesiones cerca de 10 millones de infantes a nivel mundial [18].

Actualmente en México, la mayoría de los estados de la república no cuentan con sanciones para conductores que no cuenten con sistemas de retención de infantil (SRI) dentro de su vehículo [19], únicamente cuentan con medidas preventivas en los vehículos como señalizaciones, las cuales pueden reducir en su magnitud y severidad del accidente [12]. La idea de crear una alternativa con Inteligencia Artificial en la prevención de infantes en accidentes automovilísticos es porque, Anita Schojoll señala que las maquinas tiene un margen de error del 3% y los humanos es del 5%. Dicho 2 % de diferencia justifica el uso de la inteligencia Artificial en materia de seguridad y prevención de accidentes[20]. Por lo que, la presente investigación plantea la integración de la inteligencia artificial a los vehículos automotrices, mediante la implementación de un sistema de visión computacional para la detección de infantes en el asiento de copiloto con el objetivo de la disminución de muertos y heridos graves en los accidentes automovilísticos.

1.4. Hipótesis

Mediante el uso de una cámara angular y redes neuronales convolucionales profundas es posible generar un modelo que permita identificar y clasificar un infante en el asiento del copiloto de un vehículo.

1.5. Objetivo General

Detectar infantes en el asiento del copiloto en automóviles mediante la implementación de un sistema de visión computacional y redes neuronales convolucionales.

1.5.1. Objetivos Específicos

- Establecer un protocolo de recolección de datos para menores de edad.
- Recolectar videos con distintos casos de estudio.
- Procesar los videos para cada caso de estudio.
- Generar un modelo mediante el reentrenamiento de la arquitectura GoogleNet que sea capaz de detectar la presencia de infantes en el asiento del copiloto.
- Evaluar el desempeño del modelo generado.

Capítulo 2. Marco Teórico

2.1. Descripción de Teorías Base

2.1.1. Procesamiento digital de imágenes

El Procesamiento Digital de Imágenes (PDI) es un área de la ingeniería que manipula y analiza la información contenida en una imagen digital por medio de un procesador. El procesamiento digital de imágenes es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar su calidad, mientras que el análisis de imágenes incluye aquellas técnicas cuyo principal objetivo es facilitar la búsqueda e interpretación de información contenida en ellas.[21]

Para llevar a cabo el procesamiento digital de imágenes es necesario adquirir la imagen la cual es una representación, sin parecido, o imitación de un objeto o cosa, que contiene información descriptiva acerca del objeto que representa, la adquisición de una imagen se lleva a cabo por medio de una cámara fotográfica o de video con transductores que mediante la manipulación de la luz o de alguna forma de radiación que este emitida o reflejada por los cuerpos forma una representación del objeto [22]. Posteriormente esa información pasa a ser digitalizada en forman de arreglos en una matriz con vectores ordenados de modos específicos donde se crea un mapa de la imagen haciendo una división del espacio a modo de puntos llamados pixeles. la representación numérica puede ser determinada como una función bidimensional $f(x,y)$, donde x y y son las coordenadas espaciales (plano), y la amplitud de f en cualquier par de coordenadas (x,y) es llamada intensidad. Se pueden encontrar en escala a grises con valores escalares de 0 a 255, mientras que las imágenes RGB se encuentran en el mismo rango solo que en 3 canales rojo, verde y azul. Cada pixel lleva asociada información referente al entorno capturado. La matriz suele ser almacenada en alguno de los siguientes formatos .jpg, .png, .bmp, etc. Para posteriormente ser tratada por alguna técnica de procesamiento de imágenes [23].

Se consideran tres niveles para el procesamiento digital de imágenes, la visión de bajo nivel involucra los procesos que son primarios que no requieren ningún tipo de inteligencia. La visión de nivel intermedio tiene asociado aquellos procesos que extraen características y etiquetas de la imagen, es decir involucra la segmentación, descripción y reconocimiento. La visión de alto nivel está enfocado al reconocimiento de patrones y la interpretación de escenas, infiriendo información acerca de los objetos presentes en la imagen[24]. Algunas aplicaciones de PDI son, análisis de imágenes satelitales, análisis de imágenes médicas, mediciones visuales industriales [25].

La mayoría de las técnicas de visión computacional involucran un procesado de las imágenes mediante la caracterización de la escena la cual se quiera estudiar usando alguna clase de descriptor, el cual típicamente trata de representar matemáticamente lo que se está ocurriendo en dicha imagen, un ejemplo de esto es el procesado que sufren las imágenes para mejorar la visualización, como lo es el ajuste del contraste mediante la ecualización del histograma de una imagen [26], en el cual se trata de mejorar la visualización al modificar la distribución de las intensidades de la imagen, y así mejorar la visualización.

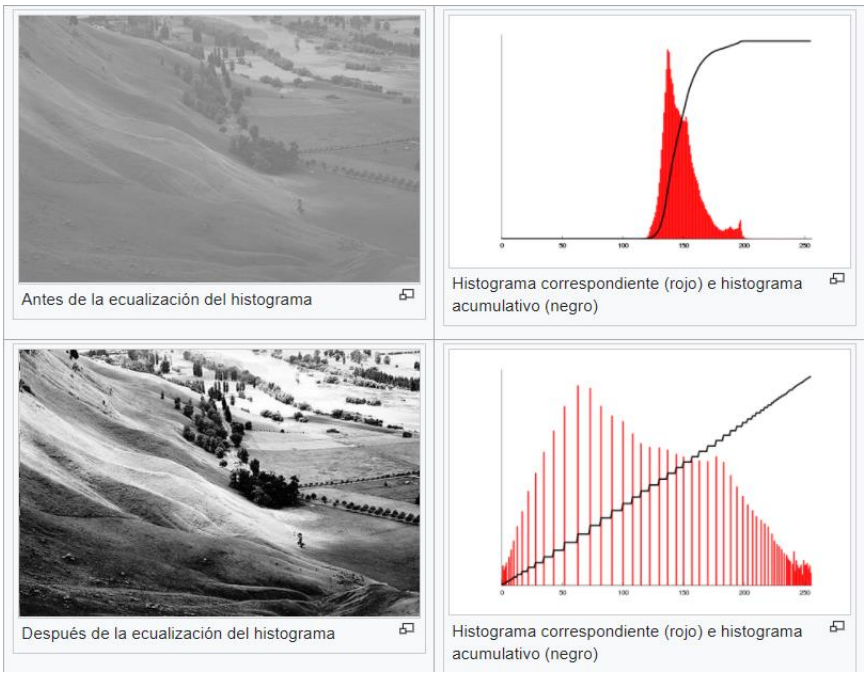


Figura 2.1 Ecualización del histograma a una imagen[26]

Típicamente este proceso de modificación de los valores de la imagen mediante la aplicación de un proceso matemático, se le llama filtrado o filtro de imagen, sin embargo, dichas técnicas son muy complejas para poder ser adaptadas a un ambiente tan dinámico como lo es el interior de un automóvil, en donde no se tiene control sobre el tipo de ropa que viste el ocupante, altura, color de piel, condiciones de luz, complexión física etc. Por lo que desarrollar técnicas (filtros de PDI), sería una labor muy compleja de atacar, es por eso que se propone el uso de inteligencia artificial como metodología para la detección de infantes sin la necesidad de desarrollar o implementar filtros de PDI complejos.

2.1.2. Inteligencia Artificial

La Inteligencia Artificial (IA) es un área de la ciencia de gran interés por ser un área multidisciplinaria donde se realizan sistemas que tratan de hacer tareas y resolver problemas como lo hace un humano, así mismo se trata de simular de manera artificial las formas del pensamiento y como trabaja el cerebro para tomar decisiones. Dentro de los sistemas de inteligencia artificial se encuentra los sistemas de aprendizaje supervisado y no supervisado[27].

Aprendizaje no supervisado: en el aprendizaje no supervisado, el conjunto de entrenamiento consta de entradas sin etiquetar, es decir, de entradas sin ninguna salida deseada asignada. El aprendizaje no supervisado generalmente tiene como objetivo intentar encontrar algún tipo de organización que simplifique el análisis. Tienen un carácter exploratorio.

En el aprendizaje supervisado son los algoritmos trabajan con datos “etiquetados”, intentado encontrar una función que, dadas las variables de entrada, les asigne la etiqueta de salida adecuada. El algoritmo se entrena con una base de datos y así “aprende” a asignar la etiqueta de salida adecuada a un nuevo valor, es decir, predice el valor de salida[28].

Los algoritmos de clasificación se usan cuando el resultado deseado es una etiqueta, en la actualidad se encuentran una variedad de algoritmos para clasificación como lo son los algoritmos de agrupamiento utilizan una serie criterios como la distancia entre los vectores de entrada para determinar a qué clase

pertenecen. Ejemplos de algoritmos de agrupamiento son: K-means, K-medoids. Otros algoritmos desarrollados por la inteligencia artificial son: el Análisis Discriminantes y los K vecinos próximos, y los Árboles de Decisión, las Máquinas Soporte Vector, Redes Neuronales.

Desde los años 60's, uno de los algoritmos más robustos que recientemente ha generado popularidad son las Redes neuronales artificiales. Estas son redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico[29]. El principal objetivo de este modelo es la construcción de sistemas capaces de presentar un cierto comportamiento inteligente. Esto implica la capacidad de aprender a realizar una determinada tarea. Una red neuronal está compuesta por varias partes como lo son las entradas, nodos, sinapsis y salidas. Las entradas reciben los datos o parámetros que le permiten decidir a la neurona se estará activa o no, normalmente se presentan como $X_1, X_2, \dots, X_n$. los pesos sinápticos son los que representan la memoria de la red y se define la fuerza de una conexión sináptica entre dos neuronas, pueden tomar valores positivos, negativos o cero, normalmente se presentan como $W_1, W_2, \dots, W_n$. Un sumador es el que se encarga de sumar todas las entradas mutiladas por las respectivas sinapsis. Las salidas de una neurona pueden ser clasificadas en dos grandes grupos, binarias o continuas. Las neuronas binarias, sólo admiten dos valores posibles. En general en este tipo de neurona se utilizan los siguientes dos alfabetos {0,1} o {-1,1}. Por su parte, las neuronas continuas admiten valores dentro de un determinado rango, que en general suele definirse como [-1, 1].[30] En la figura 2.1 se muestra él es quema de la red neuronal.

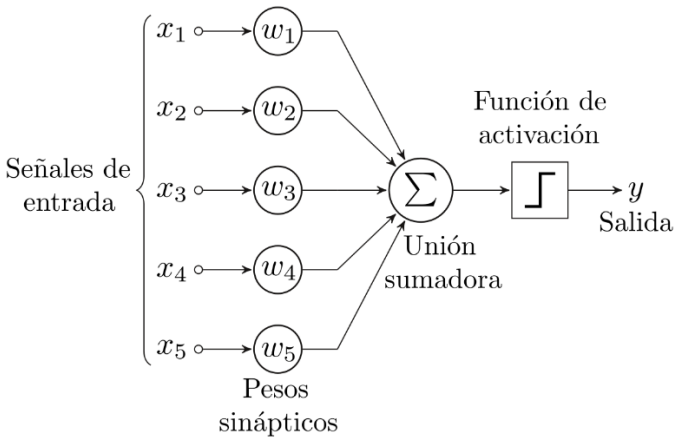


Figura 2.2Esquema de red neuronal [28].

La ecuación 2.1 muestra la representación matemática de una red neuronal.

$$x = \sum_{i=1}^{k} X_i * W_i \qquad (2.1)$$

En la figura 2.2 se muestras las funciones de activación de las redes neuronales.

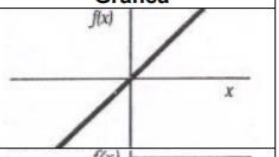
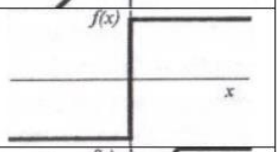
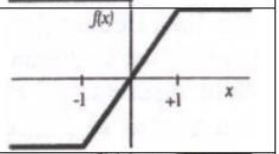
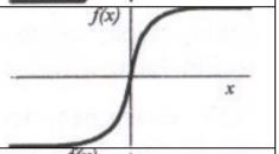
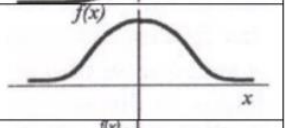
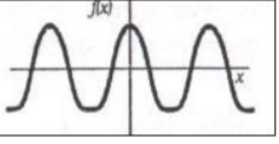
	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, \infty]$	
Escalón	$y = \begin{cases} 1, & si\ x \geq 0 \\ 0, & si\ x < 0 \end{cases}$ $y = \begin{cases} 1, & si\ x \geq 0 \\ -1, & si\ x < 0 \end{cases}$	$[0, 1]$ $[-1, 1]$	
Lineal a tramos	$y = \begin{cases} 1, & si\ x > 1 \\ x, & si\ -1 \leq x \leq 1 \\ -1, & x < -1 \end{cases}$	$[-1, 1]$	
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$ $y = \tanh(x)$	$[0, 1]$ $[-1, 1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0, 1]$	
Sinusoidal	$y = A\sin(wx + \varphi)$	$[-1, 1]$	

Figura 2.3 Funciones de activación de una red neuronal [27].

Se le denomina topología a la estructura o patrón de conexionado de una red neuronal. En una red neuronal artificial los nodos se conectan por medio de sinapsis. Estas conexiones sinápticas son direccionales, es decir, la información solamente puede propagarse en un único sentido (desde la neurona pre sináptica a la pos-sináptica). En general las neuronas se suelen agrupar en unidades estructurales que denominaremos capas. El conjunto de una o más capas constituye la red neuronal [31].

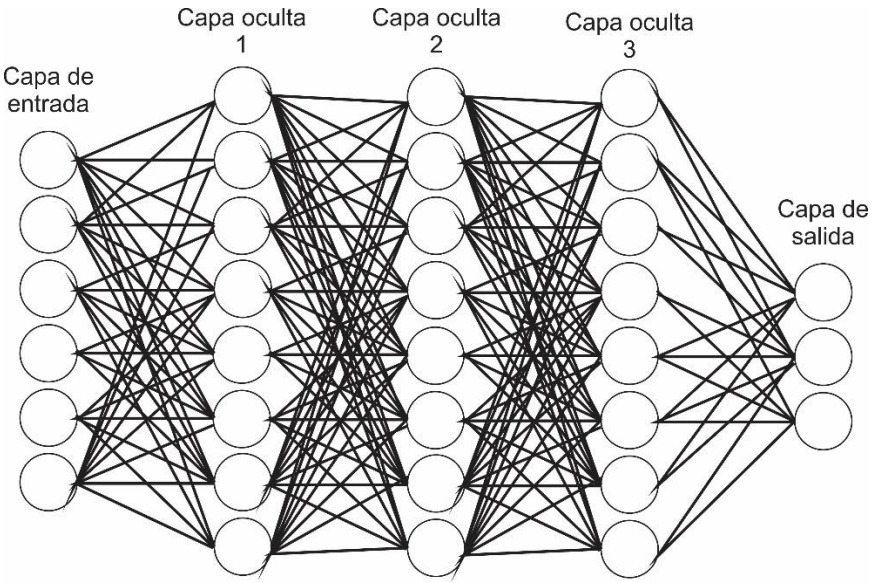


Figura 2.4 Topología de una Red Neuronal

Una vez seleccionada el tipo de neurona artificial que se utilizará en una red neuronal y determinada su topología es necesario entrenarla para que la red pueda ser utilizada. Partiendo de un conjunto de pesos sinápticos aleatorio, el proceso de aprendizaje busca un conjunto de pesos que permitan a la red desarrollar correctamente una determinada tarea. Durante el proceso de aprendizaje se va refinando iterativamente la solución hasta alcanzar un nivel de operación suficientemente bueno.

Aprendizaje supervisado. Se presenta a la red un conjunto de patrones de entrada junto con la salida esperada. Los pesos se van modificando de manera proporcional al error que se produce entre la salida real de la red y la salida esperada.

Aprendizaje no supervisado. Se presenta a la red un conjunto de patrones de entrada. No hay información disponible sobre la salida esperada. El proceso de entrenamiento en este caso deberá ajustar sus pesos en base a la correlación existente entre los datos de entrada.

Aprendizaje por refuerzo. Este tipo de aprendizaje se ubica entre medio de los dos anteriores. Se le presenta a la red un conjunto de patrones de entrada y se le indica a la red si la salida obtenida es o no correcta. Sin embargo, no se le proporciona el valor de la salida esperada. Este tipo de aprendizaje es muy útil en aquellos casos en que se desconoce cuál es la salida exacta que debe proporcionar la red.[32]

Una red neuronal profunda (DNN) es una red neuronal artificial (ANN) con varias capas ocultas entre las capas de entrada y salida, aumenta la complejidad de la red y nos permite modelar funciones que son más complicadas, sin embargo, esto aumenta el costo computacional. Estas redes neuronales se basan en un conjunto de capas ocultas, en su mayoría no lineales [33]. En las redes neuronales la primera capa de la red neuronal procesa una entrada de datos y la pasa a la siguiente capa como salida, este proceso se va repitiendo sucesivamente hasta completar la totalidad de las capas de la red neuronal. Estas redes terminan en una capa de salida: un clasificador logístico que asigna una probabilidad a un resultado o etiqueta en particular [34].

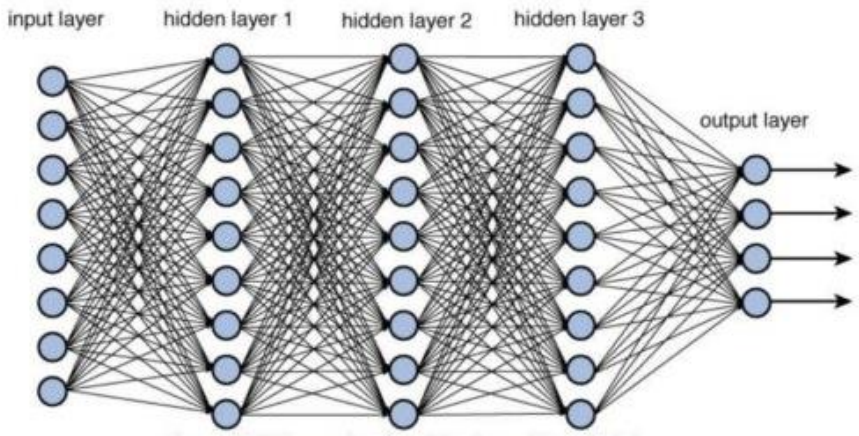


Figura 2.5 Red neuronal profunda [33].

El objetivo del entrenamiento es hacer que el costo del entrenamiento sea lo más pequeño posible a través de millones de ejemplos de entrenamiento. Para hacer esto, la red ajusta los pesos y sesgos hasta que la predicción coincida con el resultado correcto.

Las Redes Neuronales Convolucionales (CNN), son un tipo de Red Neuronal Artificial de aprendizaje automático que imita la estructura de la corteza visual primaria (V1) del cerebro con conexiones y operaciones matemáticas, donde se estrena para identificar y clasificar directamente desde imágenes, video, texto, sonido o también para sintetizar señales temporales. Son utilizadas para encontrar patrones en imágenes de gran tamaño, reconocer objetos o rostros [35].

Están diseñadas para trabajar con imágenes, tomando estas como input, asignándole importancias (pesos) a ciertos elementos en la imagen para así poder diferenciar unos de otros. Este es uno de los principales algoritmos que ha contribuido en el desarrollo y perfeccionamiento del campo de Visión por computadora. Las redes convolucionales contienen varias hidden layers, donde las primeras puedan detectar líneas, curvas y así se van especializando hasta poder reconocer formas complejas como un rostro, siluetas, etc [36].

Dentro de las redes neuronales convolucionales tienen un inconveniente, pues no existe un patrón definido para establecer el número de capas y tampoco las conexiones entre ellas. Actualmente, un enfoque para este problema común es usar las estructuras que ya se ha demostrado su efectividad y precisión en la clasificación de imágenes como lo son: LeNet, AlexNet, VGG, ResNet, DenseNet. Dentro de ellas GoogleNet que es una de las más poderosas por su exactitud y bajo costo computacional, la cual se abordará en la siguiente sección.

2.1.3. GoogleNet.

La arquitectura $inception$ cuenta con capas iniciales de convolución simple, son las encargadas de realizar el proceso de PDI, en el cual en esas capas la red neuronal realizar la de la reducción de dimensiones, resaltado y detección der bordes, o detectar bordes o enfocar la imagen, es decir, estas capas engloban el proceso que típicamente se tendría que realizar mediante el filtrado de imágenes,

lo cual acelera la implementación de estas metodologías así mismo permite una mejor abstracción de la información.

La convolución es una operación matemática que transforma dos señales $(f\ y\ g)$ en una tercer señal. La convolución matricial aplica un filtro o kernel, este consiste en una matriz con algunos valores establecidos conocidos como pesos, que recorre toda la imagen para transformar y reducir las dimensiones de la imagen original. Cada imagen es definida como una función bidimensional $z = F(x, y)$ donde x e y representan las coordenadas espaciales y z es el valor de la intensidad de la imagen en el punto (x, y). Las variaciones cambian según el contenido de bits de una imagen. Los rangos de las intensidades en una imagen varían de un valor de 0 que pertenece al color negro y a 255 que es el color blanco. Las imágenes digitales a color trabajan con tres canales R, G y B que proporcionan las intensidades correspondientes a los colores primarios rojo, verde y azul.

La convolución de matrices se muestra en la ecuación 2.2.

$$D(x, y) = \sum_{i=-k}^{k} \sum_{-p}^{p} M(i, j) \cdot F(x + i, y + j) \qquad (2.2)$$

Dada una matriz $M[-k \ldots k, -p \ldots p]$ definida como máscara o kernel y una matriz $F(x, y)$ como imagen de entrada, la convolución de las matrices F y M se denota como una nueva matriz $D = F * M$.

GoogleNet, utiliza la convolución 1X1, como módulo de reducción de dimensiones para disminuir el cálculo computacional, además ayuda a reducir el tamaño del modelo y en consecuencia evitar el problema de sobreajuste. Contiene Max-Poling que es una transformación no lineal dando como resultado un submuestreo espacial de la matriz.

Módulo de inicio.

La figura 2.5 muestra la arquitectura del módulo de inicio de $inception,$ el cual se encarga de reducir la cantidad de operaciones. Cuando una imagen llega, el módulo prueba sobre diferentes tamaños de convoluciones y agrupaciones

máximas. Posteriormente, se extraen diferentes tipos de características para generar un filtro concatenado.

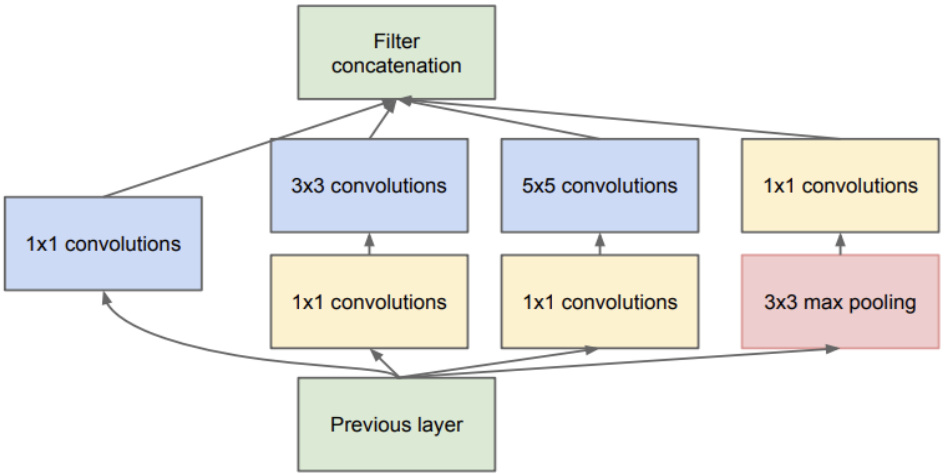


Figura 2.6 Módulo de Inicio

Arquitectura.

La figura 2.6 muestra la arquitectura de GoogleNet, cuenta con 22 capas de profundidad en donde las capas iniciales son de convolución simple, seguido de varios bloques de agrupación máxima (Max-Pool), y finalmente por capas de sofmax. Esta arquitectura cuenta con 99.18% de exactitud [37]

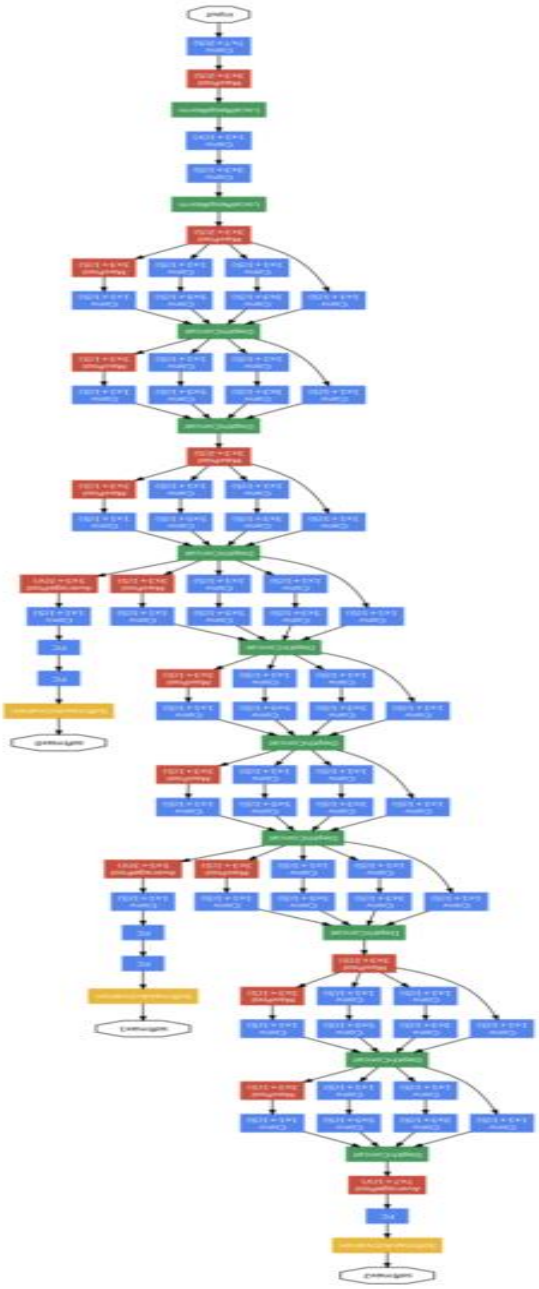


Figura 2.7 Arquitectura de GoogleNet [37]

A continuación, se detallan los parámetros de cada capa en la figura 2.7.

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Figura 2.8 Parámetros de la red GoogleNet.

Capa clasificadora

En la capa de clasificación softmax también conocida como función exponencial normalizada, estima la probabilidad de una observación que pertenece a una determinada clase. Se representa por la ecuación 2.3, esta función es aplicada sobre cada elemento zi del vector de entrada z y lo normaliza dividiéndolo por la suma todos sus exponenciales. Esta normalización asegura que la suma de los componentes del vector de salida $\sigma(z)$ sea igual a uno.

$$\sigma(z)i = \frac{e^{zi}}{\sum_{i=1}^{k} e^{zj}} \qquad (2.3)$$

Las redes neuronales convolucionales presentan varias desventajas es la complejidad de aprendizaje para grandes tareas, cuantas más cosas se necesiten

que aprenda una red, más complicado será enseñarle, el tiempo de aprendizaje es otro factor que influye, si se busca que sean sumamente precisa, se deberá invertir más tiempo en lograr que la red converja a valores de pesos que representen lo que se quiera enseñar [38]. Es por ello que se implementan técnicas como lo es el Transfer learning, se describe en el siguiente apartado.

Transfer learning

Esta técnica es una de las más importantes en Deep Learning, consiste en utilizar el conocimiento adquirido mientras resuelve un problema en particular, el objetivo de utilizar esta técnica radica en ahorrar el tiempo de entrenamiento de una red neuronal desde cero, que implica tener una gran cantidad de datos o imágenes, esto podría requerir demasiado tiempo (semanas o meses) y poder computacional [39].

La transferencia de conocimiento se divide en tres sub-secciones: transferencia de conocimiento inductiva, transferencia de conocimiento transductivo y transferencia de conocimiento no supervisada.

La transferencia de conocimiento inductiva. Es un escenario similar al escenario de aprendizaje multitarea. Sin embargo, el aprendizaje transferencia inductiva, sólo tiene por objetivo lograr un alto rendimiento en la tarea de destino mediante la transferencia de conocimiento a partir de la tarea fuente, mientras se trata de aprendizaje multitarea para aprender la tarea de destino y la fuente de forma simultánea, donde los datos etiquetados en el dominio de origen no están disponibles.

La transferencia de conocimiento transductivo. Es donde las tareas de destino son el mismo, mientras que los dominios de origen y de destino son diferentes. No hay datos etiquetados en el dominio de destino, mientras que una gran cantidad de datos etiquetados en el dominio de origen están disponibles. Además, de acuerdo a las diferentes situaciones entre los dominios de origen y de destino.

En la transferencia de conocimiento no supervisado. La tarea de destino es diferente, pero relacionada con la tarea fuente. Sin embargo, la transferencia de conocimiento no supervisada se centra en la solución de tareas de aprendizaje no

supervisado en el dominio de destino, tales como la agrupación, la reducción de dimensionalidad y la estimación de la densidad. En este caso, no hay datos disponibles etiquetados en ambos dominios de origen y de destino en el entrenamiento [40].

Según Yang y Pan [41], la transferencia de conocimiento (TL, por sus siglas en inglés) se define como: "*Dado un dominio fuente D_S y una tarea de aprendizaje T_S, un dominio de destino D_T y una tarea de aprendizaje T_T, el T_L tiene como objetivo ayudar a mejorar el aprendizaje de la función predictiva $F_T(.)$ en D_T utilizando el conocimiento en D_S y T_S, donde $D_S \neq D_T$ o $T_S \neq T_T$.*

El proceso de transferencia del conocimiento se muestra en la figura 2.8, consiste en transferir el conocimiento de algunas tareas previas a una tarea de destino cuando éste tiene un menor número de datos de entrenamiento de alta calidad.

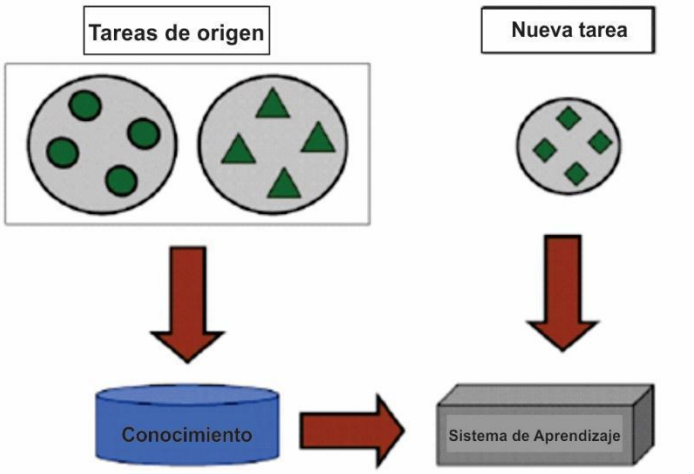


Figura 2.9 Proceso de transferencia del conocimiento.

En algunas situaciones, cuando el dominio de origen y el dominio de destino no están relacionados entre sí, la transferencia de fuerza bruta puede ser no exitosas. En el peor de los casos, puede incluso dañar el rendimiento del aprendizaje en el dominio de destino, una situación que a menudo se refiere como transferencia negativa. La forma de evitar la transferencia negativa es un problema abierto importante que está atrayendo cada vez más atención en el futuro.

2.2. Principales estudios relacionados

En la actualidad, se han propuesto distintos sistemas de Detección de pasajeros para la prevención de lesiones en accidentes automovilísticos. Estos sistemas se integran a base de sensores o cámaras, y realizan una clasificación tomando de base de características fisiológicas, por ejemplo, peso y altura. A continuación, se describe algunos trabajos relacionados identificados en la literatura.

Vehicle's Interior Presence Detection and Notification System

En el trabajo de Jóni Ramiro [42] se diseñó un sistema de bajo costo que incorpora un sensor de movimiento y sistema de visión computacional con redes neuronales profundas y la arquitectura AlexNet. Con el fin de detectar, notificar al usuario, para la reducción del número de accidentes de infantes en vehículos. Obteniendo como resultado una precisión de 81.7% y un área bajo la curva de 83.4%.

Unattended children in cars – radiofrequency-based detection to reduce heat stroke fatalities

El artículo de J.Goncalves [43] desarrollo un sistema de detección de infantes que emite señales en la banda ISM de 24 GHz y evalúa la señal reflejada. El sistema es lo suficiente sensible para detectar los pequeños movimientos respiratorios de un bebé dormido, e incluso es capaz de detectar al niño en circunstancias difíciles (ruido) en vehículos. Con el objetivo de reducir los accidentes de infantes en automóviles. Obteniendo como resultado una tasa de 82% de precisión.

Machine Learning-Based Human Recognition Scheme Using a Doppler Radar Sensor for In-Vehicle Applications

El artículo de Eugin Hyun [44] propone un esquema de detección de pasajeros basado en el espectro Doppler para un CW Sensor de radar (onda continua) en aplicaciones de vehículos. Con el objetivo de detectar humanos o infantes en un vehículo. La precisión fue de 70,7% de la clasificación y la tasa de error de 29.3% dela clasificación utilizando el algoritmo propuesto para un humano con o sin movimiento.

3-D Vision Technology for Occupant Detection and Classification

El trabajo de Pandu R. Rao Devarakota 2005 [45] describe un sistema de visión 3-D basada en sensores 3-D utilizados para la detección, clasificación y posición de los ocupantes en un automóvil, con la cual se busca tener un control de la inflación de airbags para la reducción de lesiones. llevando pruebas con infantes, asientos y adultos, obteniendo como resultados una tasa de clasificación global de 95% en la precisión.

iBolt Technology – A Weight Sensing System for Advanced Passenger Safety

En el año 2006, la empresa Bosch desarrolló un módulo de control llamado "iBolt" integrado, se comunica a través de un sistema bus con la electrónica central de control del airbag que, debido a la medición exacta de cuatro puntos, recibe más información que la simple clasificación del peso. El sistema "iBolt" evalúa los cuatro valores de medición y puede identificar la posición del asiento y sus modificaciones. Con información adicional, la activación del airbag se puede adaptar con generadores de gas de dos o más niveles a la posición del cuerpo del copiloto [46].

Advanced Occupant Detection System: Detection of Human Vital Signs by Seat-Embedded Ferroelectric Film Sensors and by Vibration Analysis

El estudio de Laurent Federspiel 2005 [47] en la detección de los seres humanos basado en un sensor de polímero electro-responsive encajado en el cojín del asiento y capaz de capturar los signos vitales humanos en escenarios de vehículos, para la clasificación de seres humanos u objetos utilizando la discriminación basada en el conocimiento. Obteniendo como resultados una precisión del 75% en infantes hasta 3 años de edad.

Occupant Classification System for Automotive Airbag Suppression

En el trabajo de Farmer 2003 [48] se describe un sistema de clasificación de ocupantes con tres clases: infantes, adultos y asiento vacío. Basado en visión artificial utilizando una cámara en escala de grises para detectar los bordes y extraer las características combinando un algoritmo de clasificación K-means que proporciona una tasa global del 95% en una base de datos de 21,000 imágenes.

Vision Based Occupant Detection System by Monocular 3D Surface Reconstruction

El artículo de Yoon 2004 [49] presenta un sistema con una cámara monocular basada en un sistema de clasificación de objetos para el despliegue del airbag del vehículo. Las imágenes son iluminadas desde diferentes direcciones para, posteriormente ser secuencializadas por el método Photometric Stereo y de esta manera, recuperar la forma tridimensional del objeto definido por un vector de 29 características para entrenar una red neuronal que clasifique en 2 clases niño o adulto. logrando obtener una precisión de clasificación del 98.9% en una base de datos de más de 84,000 imágenes.

2.3. Comparación entre los trabajos relacionados y la propuesta de investigación

A continuación, la tabla 2.1 compara los trabajos relacionados donde se muestra las diferentes técnicas y los resultados obtenidos en el área de clasificación de infantes.

Autores	Método	Métricas	Resultado
Jóni Ramiro - 2018	Cámara y sensor de movimiento	Precisión, Área bajo la curva	81.7%
J.Goncalves - 2017	Sensor	Precisión	82%
Eugin Hyun - 2020	Radar	Precisión	70.7%
Bosch - 2006	Módulo iBolt	Presión	95%
Pandu R. Rao Devarakota - 2005	Sensores 3D	Precisión, Matriz de confucin	95%
Laurent Federspiel - 2005	Sensor de polímero electro-responsive	Precisión	75%
Farmer - 2003	cámara en escala de grises	Precisión	95%

Yoon - 2004	Cámara monocular	Precisión	98.9%

Tabla 2.1 Resumen de métodos y comparación de estudios relacionados con la detección e infantes en vehículos.

En la investigación y búsqueda de los trabajos descritos y comparados anteriormente, se encontraron pocos trabajos enfocados a la detención de infantes en el área de copilo, es por ello que algunos de los trabajos relacionados presentan fechas posteriores al 2016, lo cual indica que es un área en el que aún no se han explorado el tipo de técnicas propuestas en esta tesis.

Capítulo 3. Método y propuesta de investigación.

3.1. Modelo de investigación.

La metodología propuesta para la Detección de niños en la zona de copiloto en vehículos, se encuentra dividida en cinco etapas, como se muestra en la Figura 2.1. La primera etapa corresponde a la adquisición de video, en donde al interior del vehículo se realiza la grabación de la zona interés. La siguiente etapa consiste en la eliminación de los metadatos presentes en el video capturado transformando el formato origen de GPMF a MP4. Posteriormente, el video transformado es procesado para sustraer un conjunto de imágenes. Este conjunto de imágenes se utiliza como entrada para el reentrenamiento de la red neuronal convolucional GoogleNet. Finalmente, el modelo resultante es evaluado sobre una prueba a ciegas.

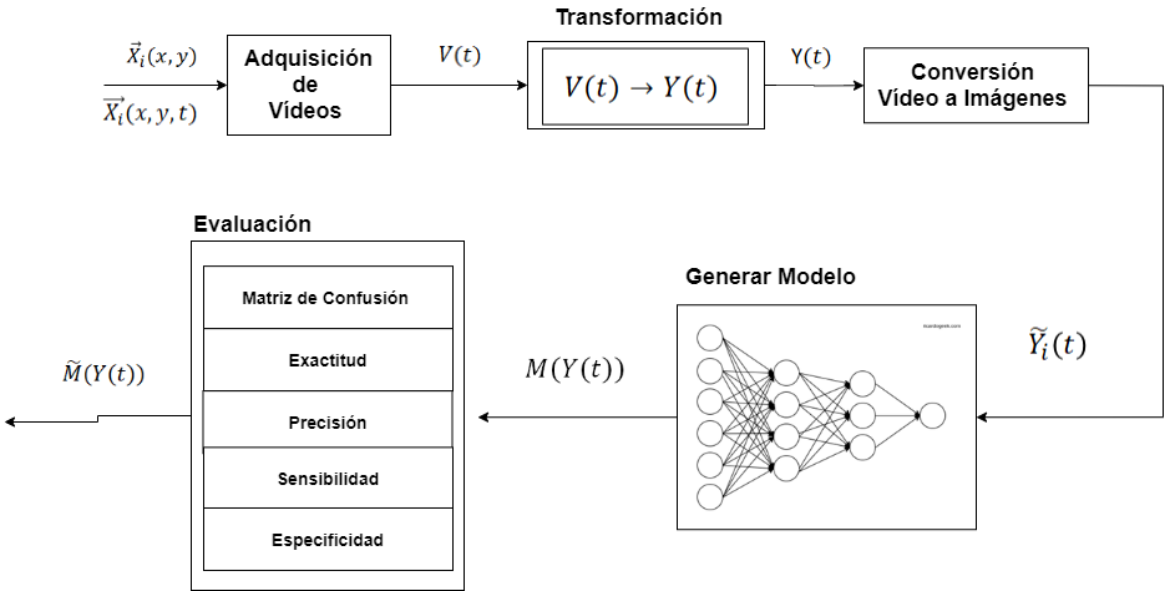


Figura 3.1 Metodología para la detección de infantes como copilotos.

3.2. Descripción de la propuesta.

3.2.1. Adquisición de videos.

Esta etapa consiste en la grabación de la zona de copiloto del vehículo de prueba. La adquisición de los videos se realizó con una cámara GoPro HERO 5 Session. La cual ofrece distintas cualidades a configurar como: Campos de Visión FOV, resolución de captura, frecuencia de captura en fotogramas por segundo (FPS). La configuración de captura que se utilizó fue de: FOV (Gran Angular), resolución de 1980 × 1080 píxeles, frecuencia de 24 FPS.

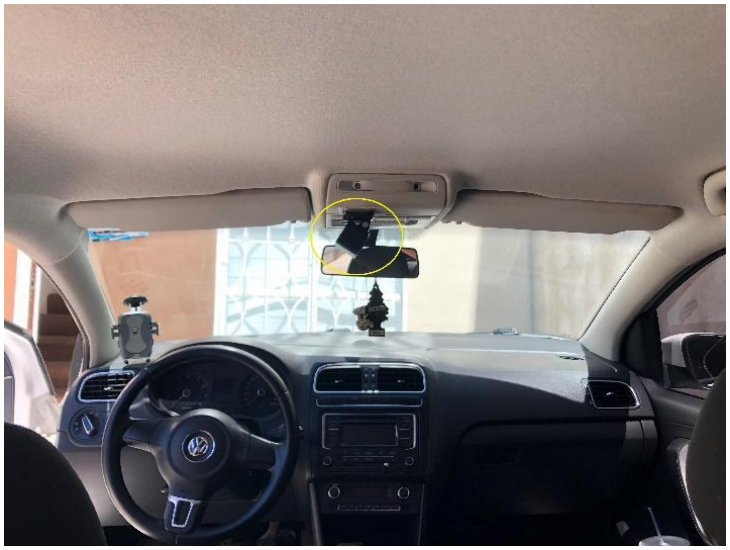


Figura 3.2 Ubicación de la cámara en el automóvil.

La captura de las imágenes se conforma por tres clases: Infante, Personas y Objetos, cada imagen se representa por:

$$\overrightarrow{X_i}(x, y) = V(t)$$

Donde el vector de entrada está representado por: $X_i \rightarrow (x, y)$ donde i representa el i ésimo cuadro de captura, (x, y) representan la intensidad de los pixeles. V(t) representa los videos capturados de los distintos casos, donde $t = \{0, \ldots, 180\}$ es el tiempo de captura en segundos.

3.2.2. Transformación.

La etapa de transformación, consiste en cambiar el formato la matriz V(t) a mp4 con FFmpeg que es un marco multimedia, capaz de codificar, convertir,

transcodificar y filtrar cualquier archivo multimedia [50] como se muestra en la figura 2.3, con el objetivo de eliminar los metadatos que entrega el formato GoPro Metadata Format (GPMF), que son TCD que pertenece al tiempo del día y el Marco de Medianoche, MET que describe la telemetría y SOS se utiliza para recuperar archivos HERO5, obteniendo como resultante una matriz $Y(t)$ con formato MP4, con las pistas AVC que pertenece al video y AAC que es el audio.

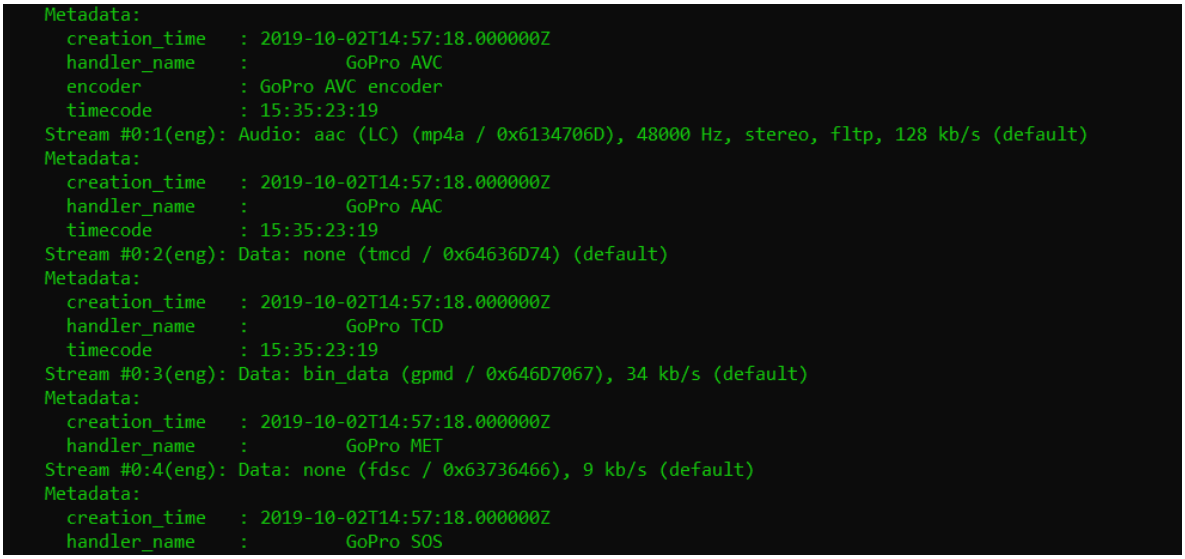


Figura 3.3 Proceso de eliminación de Metadatos.

3.2.3. Conversión de video a imágenes.

La siguiente etapa consiste en convertir los videos capturados a imágenes. Teniendo como entrada a la matriz $Y(t)$ generada anteriormente, se realiza una extracción de imágenes con la librería OpenCV (Open Source Computer Vision Library) es una librería de software enfocada en visión artificial y aprendizaje automático de código abierto [51], Cada imagen es pre-procesada para ser reducida a un tamaño de 880 x 1700 píxeles, correspondiente al área de interés, es decir, el asiento de copiloto, como se muestra en la Figura 2.4. Se obtuvo como resultado un conjunto de imágenes que se muestra en la tabla 2.1 con etiquetas según la clase a la que pertenecen.

Figura 3.4 Imagen que visualiza al área de interés en el vehículo.

Para calcular la cantidad de imágenes por video se utilizó la fórmula 2.1, donde f son los FPS y t es el tiempo del video.

$$i = f * t \tag{2.1}$$

Etiqueta	Clase	Número de imágenes
(1)	Objetos	14,310
(2)	Infantes	15,587
(3)	Personas	13,600

Tabla 2.1 Conjunto de imágenes que se tomaron en cuenta.

3.2.4. Generación del modelo.

En la etapa generación de un modelo, se utilizó la red Google CNN pre-entrenada, con una arquitectura llamada $inception$ la cual fue ganadora del ImageNet Large-Scale(ILSVRC) en el 2014, logrando proponer un nuevo estado del arte para las clasificación y detección de imágenes. Con un subconjunto de 1.2 millones de imágenes de entrenamiento, 50,000 de validación y 100,000 de prueba [37].

3.2.5. Configuración particular.

Para el propósito específico de detectar infantes en el asiento del copiloto, se utiliza una red CNN pre-entrenada por Google. La cual tiene una alta eficiencia en la clasificación de imágenes, fue entrenada con un subconjunto de 1000 clases, 1.2 millones de imágenes de entrenamiento y 50,000 de validación. Sin embargo, para el caso en particular solo busca detectar las siguientes tres clases: Infantes personas u objetos. Por lo que reentrenar dicha red, para nuestro caso nos requeriría una gran cantidad de imágenes y un alto costo computacional del cual no se cuenta. Por lo anterior, se selecciona la técnica de transferencia de conocimiento descrita en la sección 2.4.1, que consiste en reemplazar las últimas capas (capa completamente conectada, softmax y capa de salida) de la red GoogleNet (final_traing_ops) por capas reentrenadas para el problema en específico del conjunto de imágenes $Y_i(t)$. Para obtener un modelo altamente especializado denotado por $M(Y(t))$. En la figura 2.9 se muestra en proceso de reentrenamiento de la red GoogleNet.

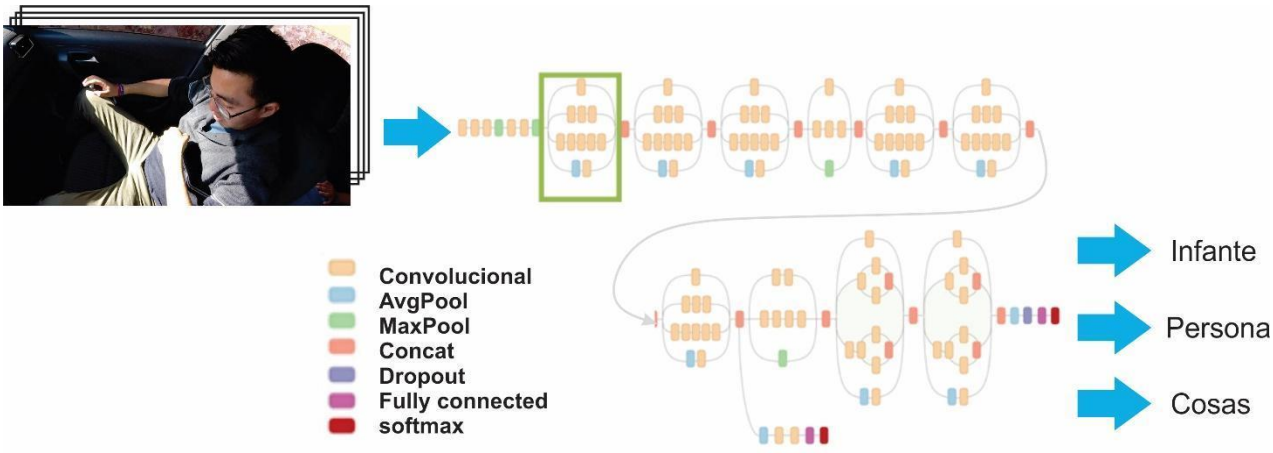


Figura 3.5 Proceso de Reentrenamiento de la red GoogleNet.

3.3. Evaluación.

En la presente etapa se describen las métricas cuantitativas que nos permiten evaluar el desempeño del modelo generado por la red neuronal para la detección de infantes.

3.3.1. Matriz de confusión.

La matriz de confusión sirve para evaluar un modelo de clasificación de infantes en vehículos donde se tiene; Un número de clases, el número de predicciones correctamente el número de clasificaciones que fueron asignadas incorrectamente para un modelo que es evaluado por una matriz de confusión se espera que la mayoría de los resultados estén en la diagonal principal, dicho de otra manera, que sean correctamente calificados ya sea VP o VN.

		Predicción	
		Positivos	**Negativos**
Observación	**Positivos**	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Figura 2.10 Matriz de confusión

En la figura 2.10 se muestra la matriz de confusión, que es un mecanismo visual para evaluar el modelo, las columnas representan el valor real de cada clase, mientras que las filas son el valor predicho por el clasificador. la diagonal principal contiene la suma de todas las predicciones correctas, asignadas por el clasificador.

A continuación, se describe cada uno de los parámetros.

Verdaderos Positivos (VP): Es el número de predicciones correctas de clase positiva (positivos reales).

Falsos Positivos (FP): Es el número de predicciones incorrectas de clase positiva (falsos positivos).

Falsos Negativos (FN): Es el número de predicciones incorrectas de clase negativa (falsos negativos).

Verdaderos Negativos (VN): Es el número de predicciones correctas de clase negativa (negativos reales).

3.3.2. Matriz de confusión multi-clase.

En la figura 2.11 se muestra la matriz de confusión multi-clase con tres clases (A, B, C), donde se exponen los datos de evaluación en una tabla. Los valores TP_A, TP_B, TP_C, representan el número de muestras clasificadas correctamente a la clase A, B, C. E_{BA} caracteriza el número de muestras de la clase A que se clasificaron incorrectamente a la clase B, es decir muestras mal clasificadas. Por lo tanto, el falso negativo en la clase A (FN_A) es la suma de E_{AB} y E_{AC} $(FN_A = E_{AB} + E_{AC})$ que indica la suma de todas las muestras de clase A que se clasificaron incorrectamente como clase B o C. Simplemente, FN de cualquier clase que se encuentre en una columna se puede calcular agregando los errores en esa clase / columna. Mientras que el falso positivo para cualquier clase predicha que se encuentra en una fila representa la suma de todos los errores en esa fila [52].

		Valor Real		
		A	B	C
Valor Predicho	A	TP_A	E_{BA}	E_{CA}
	B	E_{AB}	TP_B	E_{CB}
	C	E_{AC}	E_{BC}	TP_C

Figura 3.6 Matriz de Confusión multi-clase.

3.3.3. Exactitud.

La exactitud o Accuracy (AC) es una métrica básica que representa la proporción entre el número de predicciones correctas (tanto positivas como negativas) y el total de predicciones y se calcula mediante la ecuación 2.4. La precisión de esta métrica depende del balanceo de las clases.

$$AC = \frac{VP + VN}{VP + VN + FN + FP} \qquad (2.4)$$

3.3.4. Sensibilidad.

La sensibilidad o Recall (TP), se le conoce como la tasa de verdaderos positivos (True Positive Rate), proporciona la probabilidad de clasificar correctamente a los infantes en el vehículo. La fórmula para estimar la sensibilidad se muestra en la ecuación 2.6, donde VP son las imágenes que fueron clasificadas correctamente como infantes y FN son las imágenes de infantes que fueron clasificadas como objetos o personas.

$$Sensibilidad = \frac{VP}{VP + FN} \tag{2.6}$$

3.3.5. Especificidad.

Es una medida para estimar la tasa de verdaderos negativos (TN) tasa de Verdaderos Negativos, la cual representa la cantidad de imágenes que no fueron clasificadas como infantes correctamente, se expresa en fórmula 2.7.

$$Especificidad = \frac{TN}{TN + FP} \tag{2.7}$$

Capítulo 4. Resultados y Limitaciones

En la presente sección se describe la implementación de cada una de las etapas de la metodología llevada a cabo para generar un sistema para la detección de infantes en el asiento del copiloto en automóviles mediante el uso de visión computacional y redes neuronales convolucionales.

4.1. Carta consentimiento.

Antes de comenzar a describir cada una de las etapas de la presente metodología es importante mencionar la elaboración de una carta de consentimiento (Carta adjunta en el Anexo 1), en donde los padres y madres otorgan autorización de la participación de sus hijos durante los experimentos realizados. En la carta se informa del peligro que existe al llevar a los niños en la parte frontal del vehículo, además se describe el proyecto propuesto y los permisos que los padres y madres están cediendo de sus hijos.

Los datos de captura recabados de las cartas son: Número de participante, hora a la cual fue grabado en video, ciclo del día, edad del infante, estatura, peso y sexo. El objetivo de su consignación es tener una base de datos con la mayor cantidad de información descriptiva de cada uno de los participantes.

4.2. Adquisición de videos.

La etapa de adquisición de videos consiste en recolectar tres clases de interés: Infantes, Personas y Objetos. Para ello, se utilizó una cámara GoPro Hero 5Session y un dispositivo electrónico (iPad) que permite configurar y controlar la cámara, cabe resaltar que se pueden utilizar otro dispositivo, siempre y cuando cumpla con las necesidades de la aplicación GoPro.

La cámara fue montada en la parte media frontal del techo de un Volkswagen Vento 2014 a un ángulo de 40º enfocada a la parte del copiloto, como se muestra en la Figura 3.1.

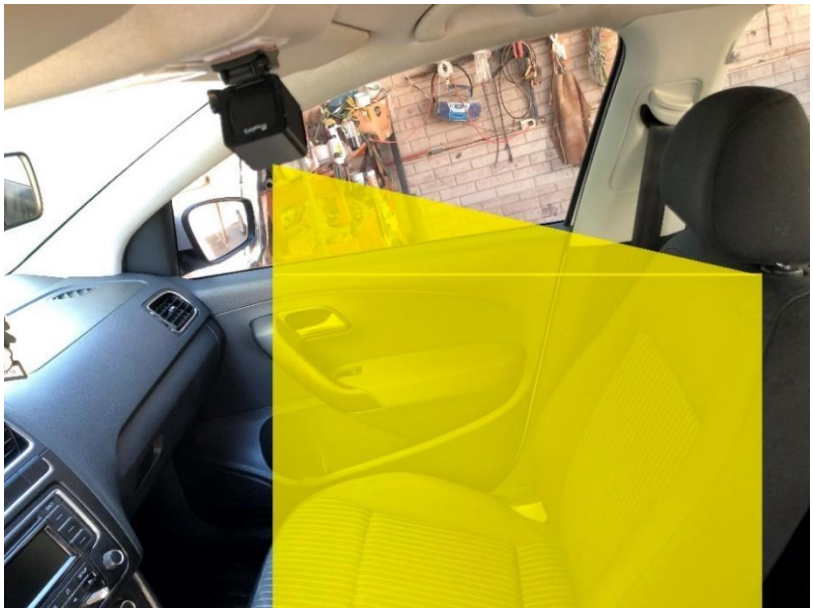


Figura 4.1 Enfoque de la cámara a zona de interés.

La configuración de la cámara consistió en los siguientes parámetros: Resolución 1080x1980 pixeles, FPS 24, campo de visión Gran Angular, estabilización del video, entre otros. En la figura 3.2 se muestra la configuración completa de la cámara en la aplicación GoPro.

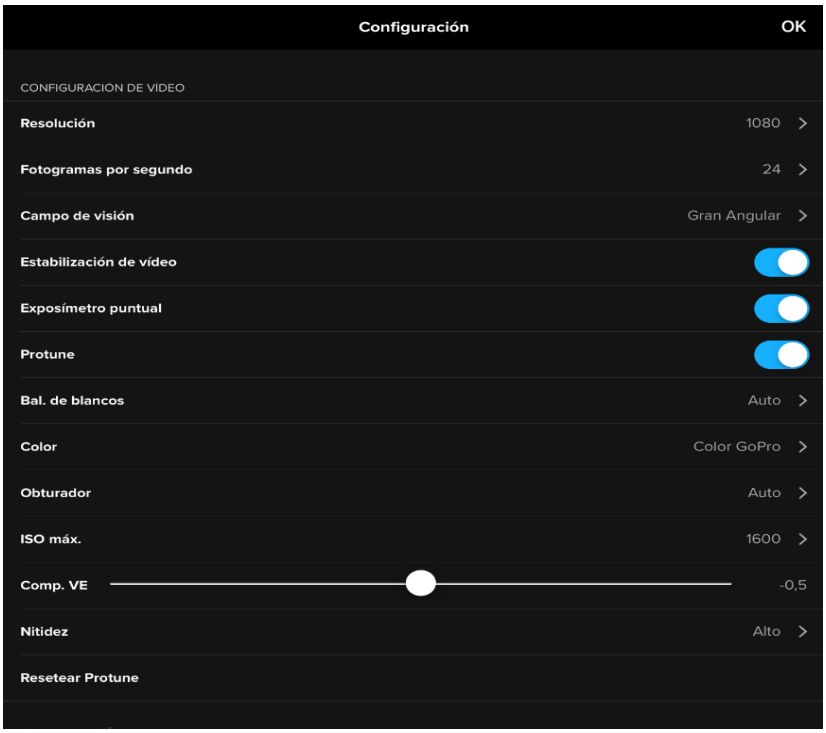


Figura 4.2 Configuración de la cámara

La captura de los videos de las clases de interés se llevó a cabo en la Universidad Autónoma de Zacatecas (UAZ) en la estancia infantil Siglo XXI, como se muestra en la Figura 3.3.

a) **Clase de interés: Infantes.**

El experimento consistió en recolectar los datos personales de cada uno de los tutores de los infantes al autorizar la carta de consentimiento. Posteriormente, se procedió a realizar los experimentos que consistieron en la colocación de los infantes en el asiento de copiloto en un vehículo estacionado.

Figura 4.3 Estancia infantil siglo XXI CASA DE LA MUJER UNIVERSITARIA.

Del conjunto de videos de infantes se obtuvo una tabla de distribución de las características que se muestra en la tabla 3.1, donde se tiene dos columnas, entrenamiento y prueba con los rangos de las edades en meses, el peso de cada infante en kilogramos, el sexo y la estatura en centímetros.

Tabla de distribución de características en casos de infantes		
	Conjunto de entrenamiento	**Conjunto de prueba**
Edad(meses)	10 - 22	30
Peso(kg)	8 - 10	9.2
Sexo	(3)M - (1)F	(0)M-(1)F
Estatura(cm)	72 - 147	100

Tabla 3.1 Tabla de distribuciones de características con infantes.

b) Clase de interés: Personas adultas

Las características recolectadas para las personas se muestran en la Tabla 3.2.

Tabla de distribución de características en casos de personas		
	Conjunto de entrenamiento	**Conjunto de prueba**
Edad(meses)	312 - 264	276
Peso(kg)	57.5 - 94	86.5
Sexo	(5)M - (0)F	(1)M-(1)F
Estatura(cm)	72 - 147	186

Tabla 3.2 Tabla de distribuciones de características con personas

c) Clase de interés: Objetos

Para la adquisición de los videos de objetos se buscó que fueran objetos ordinarios, que tuvieran el peso con el cual se activa el sensor de presencia de los vehículos. En la Tabla 3.3 se muestra los objetos que fueron utilizados para el experimento.

Tabla de objetos	
Conjunto de entrenamiento	**Conjunto de prueba**
Mochila	Portabebés
Chamara	Bolsa
Balón	Libros
Cajas	
Proyector multimedia	

Tabla 3.3 Tabla de objetos tomados en cuenta para el estudio.

Para la transformación de videos, reentrenamiento de la red GoogleNet y evaluación del modelo se utilizó una laptop Dell G7 con procesador Intel Core i7-8750h, 16GB de RAM, tarjeta gráfica NVIDIA GeForce GTX 1060, disco duro de estado sólido de 128 GB y otro disco duro rígido de 1000 GB.

4.3. Transformación de Videos.

La transformación de los videos consistió en eliminar los metadatos entregados por el formato GoPro Metadata Format (PGMF). Para la transformación del formato de los videos fue necesario ejecutar siguiente línea de código en la terminal (símbolo del sistema):

```
ffmpeg -i Video Entrada.mp4 Video_salida.mp4
```

En donde, Video_ Entrada.mp4 es el nombre del video original producido por la GoPro y Video Entrada.mp4 es el nombre del video de salida con el formato que se desea. Previamente ubicados en la carpeta o dirección donde se encuentran los videos a transformar.

Este proceso se realizó de manera manual para cada uno de los videos capturados, logrando obtener un conjunto de videos ya pre-procesados y listos para la siguiente etapa.

4.4. Conversión de videos a imágenes.

Con el objetivo de segmentar los videos a imágenes. Se desarrolló un programa en Python el cual es un lenguaje de programación orientado a objetos multiplataforma [53], con las librerías OpenCV y Numpy, que fuese capaz de cambiar los videos a imágenes. Los siguientes comandos son para instalar las librerías desde la terminal (símbolo del sistema).

```
>>pip install numpy
>>pip install opencv
```

El presente programa se tiene que importar las librerías cv2 siendo una biblioteca de visión artificial y numpy, la cual sirven para operaciones matemáticas y de matrices. Estas conteniendo las funciones necesarias para desarrollar nuestro objetivo específico de esta etapa. En la figura 3.1 se muestran los elementos principales donde se especifica el nombre del video, el formato del video y con ello creamos una carpeta acorde a la clase que pertenezca al video.

```
import numpy as np
import cv2, os

archivo = '7.1'
extencion = '.mp4'
clase='personas_test'
if not os.path.exists(clase):
    os.makedirs(clase)
cap = cv2.VideoCapture(archivo + extencion)
```

Figura 4.4 Encabezado del programa para convertir el video a imágenes.

Al estar el video en función de FPS, es necesario extraer cada imagen del video. En la figura 3.2 se muestra el código encargado de leer y guardar cada frame o captura en una variable.

```
while True:
    ret, frame = cap.read()
    if ret == True:
        img = frame
```

Figura 4.5 Código para extraer las imágenes.

Una vez que se tuvo la imagen de captura, se le realizó un pre-procesamiento que consistió en recortar la imagen de 1080X1980 a 880 x 1700 píxeles, con el fin de eliminar ruido en las imágenes que servirán como entrada para la red neuronal, anulando las áreas como la ventana lateral derecha para evitar el destello de vehículos en movimiento y destellos ocasionados por el sol en el tablero. Este proceso es iterativo hasta que termina de segmentar todo el video.

```
#Para training VW vento
#               (alto)   Y(Ancho)
imagen2 = frame[200:1080,200:1980] # la imagen es 1080X1920
#cv2.imshow('recortado',imagen2)
cv2.imwrite(clase+ "/7.1%d.jpg" % count, imagen2)     # save
count += 1
```

Figura 4.6 Código para recorte de las imágenes y asignación de nombre.

Dicho proceso se realizó de manera manual para cada uno de los videos. Logrando obtener un conjunto de imágenes procesadas como se muestra en la tabla 3.4.

Tabla de imágenes			
	Entrenamiento	Prueba	Total
Infantes	15,588	3,276	18,864
Personas	13,600	4,341	17,941
Objetos	14,310	8,614	22,924
Total	43,498	16,231	59,729

Tabla 3.4 Tabla con el total de imágenes procesadas para cada clase.

4.5. Generar el modelo

En la presente etapa se realizó el re-entrenamiento de la red GoogleNet, como la cual cuenta con millones de parámetros que puede clasificar una gran cantidad de clases. Auxiliándose a cumplir con el objetivo de obtener una red neuronal convolucional profunda que sea capaz de clasificar Infantes, personas u objetos. Se utilizó como entrada el conjunto de imágenes pre-procesadas $\tilde{Y}_i(t)$ para el reentrenamiento de la última capa(final training ops) que se muestra en la Figura 3.7, que es de tipo convolucional pero que se caracteriza por utilizar únicamente filtros 1x1, 3x3 y 5x5 de manera simultánea, por lo que de esta manera consigue disminuir de forma importante el número de parámetros a calcular, como conservando su estado entrenado de todas las capas anteriores y ajustando la fundición de salida para el objetivo en particular.

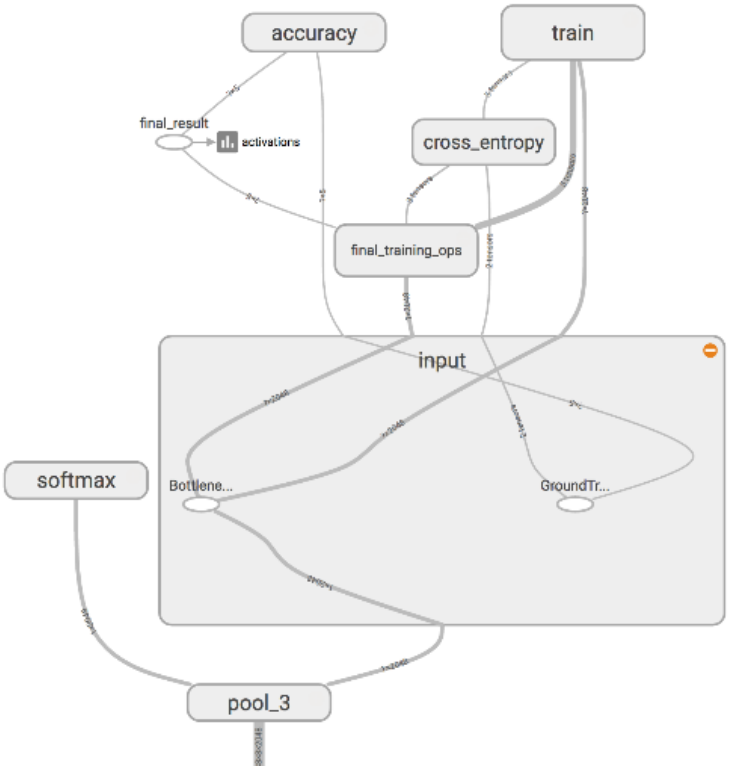


Figura 4.7 Última capa de la red GoogleNet.

Para llevar a cabo el reentrenamiento de la red GoogleNet es necesario instalar los siguientes programas: Python, Visual y Tensor Flow desde la terminal ejecutando siguientes comandos:

>>instalar Python 3.6.5 (Versión 64 Bits)

>> Microsoft Visual C++ 2015 Redistributable Update 3

>> pip install tensorflow-1.12.0-cp36-cp36m-win_amd64.whl

El proceso que se realizó fue reentrenar el grafo de la red, las direcciones de las carpetas con las clases y la cantidad de pasos o épocas que itere. Para este caso se le asignaron las direcciones de las carpetas donde se tienen las imágenes de niños personas u objetos. El siguiente comando que se muestra en la Figura 3.8 es descargar el modelo previamente entrenado y agregar una capa final y entrenarla con las fotos que previamente se procesaron.

```
python retrain.py --bottleneck_dir="D:/Proyecto Maestria/Proyecto
/tensor/tf_files/bottlenecks" --how_many_training_steps 50000
--model_dir="D:/Proyecto Maestria/Proyecto/tensor/tf_files/
inception" --output_graph="D:/Proyecto Maestria/Proyecto/tensor/
tf_files/retrained_graph.pb" --output_labels="D:/Proyecto
Maestria/Proyecto/tensor/tf_files/retrained_labels.txt"
--image_dir "D:/Proyecto Maestria/Proyecto/tensor/tf_files/
images"
```

Figura 4.8 Cogido para reentrenamiento de la red Google Net.

4.6. Predicciones

Para evaluar el rendimiento se realizaron 2 programas, el primer programa es para hacer las predicciones con nuevas imágenes y el segundo programa sirve para evaluar la precisión de la red neuronal.

a) Programa para predecir las imágenes.

El primer programa fue desarrollado en la plataforma Python, dicho programa está seccionado en 4 partes donde se cargan las librerías y los archivos que se van a necesitar para la predicción, otra etapa del programa es donde se le entrega a la

red neuronal las imágenes que va a predecir, por último, evaluamos las probabilidades entre clase y se le asigna la clase a la que pertenece.

La primera etapa del programa consiste en cargar las librerías de “Tensor Flow”, numpy, cv2, csv y sis las mismas que contienen herramientas o funciones que permiten abrir archivos editarlos, manipular grafos, hacer operaciones de matrices como el caso de las imágenes. En la Figura 3.9 se muestra donde se importan las librerías necesarias y las variables donde se aloja el grafo de la red neuronal ya entrenada.

```
import tensorflow as tf, sys
import numpy as np
import cv2
import sys, getopt
import csv

fieldnames=[]
inputpath = ''

graph_path = inputpath+'retrained_graph.pb'
labels_path = inputpath+'retrained_labels.txt'
```

Figura 4.9 Primera parte del programa para predecir las imágenes.

La segunda fase del programa consiste en abrir los archivos con la extensión .CSV como se muestra en la Figura 3.10. El primer archivo contiene los nombres de todas las imágenes a evaluar y a la clase que pertenece. El segundo archivo que se necesita es dónde se van a guardar el nombre de la imagen, la clase a la que pertenece y las portabilidades que arroja la red neuronal de pertenecer a las distintas clases.

```
with open('input.csv', newline='') as csvfile:
    reader = csv.DictReader(csvfile)

    f = open('salida_test_prob.csv','w')
    asd= ['Imagen', 'GT', 'PRE','Prob1','Prob2','Prob3']
    writer = csv.DictWriter(f, fieldnames = asd)
    writer.writeheader()
```

Figura 4.10 Abrir los archivos del programa para predecir las imágenes.

En la figura 3.11 se muestra la tercera fase del programa, misma que está enfocada a la predicción de las imágenes. Esta funciona almacenando cada una de las

imágenes escaladas a una dimensión de 300x300 píxeles y una interpolación bicúbica en un vecindario de 4X4 píxeles con el objetivo de no perder tanta información de las imágenes, siendo 300x300 la resolución máxima con la cual trabaja la red de GoogleNet. Después se le entrega al grafo o modelo entrenado la imagen y como resultado entrega las predicciones siendo la probabilidad de que pertenezca a las distintas clases.

```
frame = cv2.imread(imagenlig)

imagen2 = frame[200:1400,200:1900]

frame = cv2.resize(imagen2, (299, 299), interpolation=cv2.INTER_CUBIC)
numpy_frame = np.asarray(frame)
numpy_frame = cv2.normalize(numpy_frame.astype('float'), None, -0.5, .5, cv2.NORM_MINMAX)
numpy_frame = np.expand_dims(numpy_frame, axis=0)
predictions = sess.run(softmax_tensor, {'Mul:0': numpy_frame})
#print(predictions);
```

Figura 4.11 Tercera etapa del programa para predecir las imágenes.

En la figura 3.12 se muestra la cuarta etapa del programa. La cual consiste en evaluar las predicciones de la red neuronal para asignarlas a una clase y escribir un nuevo archivo. Para la selección o asignación de la clase a cada imagen se evalúa y compara la probabilidad de que perteneciera a la clase de Infante, Persona u Objeto y se asigna a la clase que tenga mayor valor. Con el objetivo de evaluar el rendimiento del modelo se genera un archivo CSV en el cual se guarda el nombre de la imagen, la clase a la que pertenece, la clase predicha por el modelo y las distintas probabilidades que podría obtener.

```
if ( (predictions[0][0] > predictions[0][1]) & (predictions[0][0] > predictions[0][2]) ):
    PRE=1;
    #print("cosas");#cosas
if ( (predictions[0][1] > predictions[0][0]) & (predictions[0][1] > predictions[0][2]) ):
    PRE=2;
    #print("niño");#niño
if ( (predictions[0][2] > predictions[0][0]) & (predictions[0][2] > predictions[0][1]) ):
    PRE=3;
    #print("personas");#personas


writer.writerow({'Imagen': str(IMAGEN), 'GT': str(GT), 'PRE': str(PRE), 'Prob1': str(predic
```

Figura 4.12 Cuarta etapa del programa para predecir las imágenes.

b) Programa para evaluar el rendimiento del modelo

En base al objetivo de evaluar el desempeño del modelo. Se desarrolló un programa en el lenguaje de R. Este es un entorno de software libre para el análisis de estadística y gráficos [54]. La versión de R que se utilizó fue R-3.1.0 siendo esta la versión requerida para el paquete caret, la cual incluye funciones que facilitan el uso de decenas de métodos complejos de clasificación y regresión [55].

La librería caret es una herramienta útil en R para evaluar un modelo predictivo de clasificación. Se conoce el valor esperado y el pronosticado, a partir de esto podemos obtener la matriz de confusión y las estadísticas útiles como: [56].

- Accuracy (Exactitud)
- Sensitivity (Sensibilidad)
- Specificity (Especificidad)
- IC (intervalo de confianza)
- P-Value (P-Valor)

El proceso de evaluación consta de la ejecución de tres comandos, el primero tiene como fin cargar el archivo que fue creado por el programa de predicción de imágenes, el segundo es crear una tabla con los valores a los cuales pertenece cada imagen y los valores predichos por el modelo. El tercer comando nos permite visualizar toda la información estadística ya planteada anteriormente.

```
>> salida_test_prob <- read.csv("D:/Proyecto
Maestria/Proyecto/tensor/tf_files/salida_test_prob.csv",
header=FALSE)
>> tabsal<-
table(salida_test_prob$PRE,salida_test_prob$GT)
>> print(confusionMatrix(tabsal))
```

Capítulo 5. Resultados

En el presente capítulo se exponen los resultados obtenidos en la evaluación el desempeño del modelo generado por la red neuronal, mediante las siguientes métricas: Accuracy (exactitud), Sensibilidad, Especificidad, IC (intervalo de confianza).

Para evaluar el desempeño del modelo generado por la red neuronal se realizaron 2 pruebas, la primera consiste en evaluar el desempeño del modelo entrenado con las mismas imágenes de entrenamiento como entrada y la segunda prueba consistió en evaluar con nuevas imágenes.

En la primera evaluación se tomaron en cuenta las 15588 imágenes de infantes, las 13600 imágenes de personas y las 14310 imágenes de objetos con las cuales se generó el modelo. En la tabla 4.1 podemos observar el desempeño del modelo mediante una matriz de confusión, en la matriz se puede observar una diagonal en color azul que indica la cantidad de imágenes asignadas correctamente a su clase. Los valores que están a los extremos de la diagonal representan la cantidad de imágenes asignadas a clases que no pertenecen. En la clase interés que es la de infantes se obtuvo un 0.7940 de precisión. En la clase de Objetos se obtuvo 0.8975 de precisión y en la clase de Personas se obtuvo 1 de precisión, siendo la clase con mayor precisión al momento de clasificar las imágenes.

		Real		
Predicho		Objetos	Infantes	Personas
	Objetos	12844	1	0
	Infantes	268	12378	0
	Personas	1198	3208	13600

Tabla 4.1 Matriz de confusión imágenes de entrenamiento.

En la tabla 4.2 se exponen los resultados obtenidos por el modelo en las métricas de exactitud y de intervalo de confianza (CI). Como se puede observar, el desempeño del modelo es bueno, puesto que el valor estadístico significativo del

89.25 %, lo que significa que el modelo es capaz de distinguir a un infante de una persona u objeto con solo el 10.75% de error.

Exactitud: 0.8925
95% CI : (0.8896, 0.8954)

Tabla 4.2 Exactitud y CI primera evaluación.

Tomado en cuenta la sensibilidad y la especificidad como estimador para variables estadísticas, en la tabla 4.3 se exponen los resultados obtenidos en sensibilidad y especificidad en las tres clases a evaluar. En la clase de objetos una especificidad del 100% y una sensibilidad del 89.76% significa que el modelo es muy bueno en la detección de objetos. En la clase de infantes se obtuvo una especificidad de 99.04% y sensibilidad de 79.41% también puede se puede afirmar que es bueno en la detección de infantes. En lo que podemos inferir que el modelo tiene un sesgo a las clases de Objetos y Personas.

	Objetos	Infantes	Personas
Sensibilidad	0.8976	0.7941	1.0000
Especificidad	1.0000	0.9904	0.8526

Tabla 4.3 Resultados de sensibilidad y especificidad por clase primera evaluación

Para la segunda evaluación del rendimiento se tomaron en cuenta nuevas imágenes, que incluían nuevos sujetos de prueba y nuevos objetos. De los cuales fueron 3276 imágenes de infantes, 4331 de personas y 8614 para objetos.

En la tabla 4.2 podemos observar la matriz de confusión con la cual se evaluó el rendimiento del clasificador. La diagonal azul contiene las cantidades de imágenes clasificadas correctamente a su clase. Al enfocarse a la clase de interés que es la de Infantes se puede observar que de las 3276 imágenes clasificó 3103 correctamente y 172 asignados a la clase de Personas, obteniendo un 0.9471 de precisión para la clase de infantes. las imágenes que fueron clasificadas

incorrectamente se evaluaron individualmente, en dicho análisis se observaron anomalías en la iluminación de las imágenes.

Para la clase de Personas podemos observar que clasificó correctamente a todas las imágenes. Obteniendo una precisión de 1. Por otra parte, la clase de Objetos obtuvo un 0.9757 de precisión, de las que 8405 imágenes se clasificaron correctamente y 208 asignadas a las otras dos clases, al igual que en la clase de Infantes se evaluaron individualmente y se observaron anomalías en la iluminación de las imágenes.

		Real		
Predicho		Objetos	Infantes	Personas
	Objetos	8405	0	0
	Infantes	107	3103	0
	Personas	101	172	4341

Tabla 4.4 Matriz de confusión resultados con nuevas imágenes.

En la tabla 4.5 se exponen los resultados obtenidos en la segunda prueba por el modelo en las métricas de exactitud y de intervalo de confianza (CI). Como se puede observar como el desempeño del modelo es excelente, puesto que el valor estadístico significativo del 97.66 %, lo que significa que el modelo es capaz de distinguir a un infante de una persona u objeto con 2.34% de error.

Accuracy : 0.9766
95% CI : (0.9741, 0.9789)

Tabla 4.5 Exactitud y CI Segunda evaluación.

En la tabla 4.6 se exponen los resultados obtenidos en sensibilidad y especificidad en las tres clases a evaluar de la segunda prueba. Obteniendo en la categoría de objetos una especificidad del 100% y una sensibilidad del 97.58% siendo que el modelo es muy bueno en la detección de objetos. En la clase de infantes se obtuvo

una especificidad de 99.17% y sensibilidad de 94.75%. Dados estos resultados puede afirmarse que esta clase es excelente en la detección de infantes.

	Objetos	Infantes	Personas
Sensibilidad	0.9759	0.9475	1.0000
Especificidad	1.0000	0.9917	0.9770

Tabla 4.6 Resultados de sensibilidad y especificidad por clase segunda evaluación

Para ilustrar, en la Figura 4.1 está graficado el comportamiento de la precisión en el reentrenamiento de la arquitectura GoogleNet. En la ella se observan dos líneas, la línea de color naranja representa el comportamiento del entrenamiento y la azul representa validación. Los valores del eje X representan las épocas con las cuales se entrenó la arquitectura. Los valores del eje Y son la eficiencia lograda para clasificar las imágenes, siendo 1 el valor máximo y 0 mínimo.

En la gráfica se pueden observar varios picos decrecientes que se asocian a pérdidas o malas clasificaciones con las imágenes, esto puede ser debido a que algunos factores que influyen para la buena clasificación son; Saturación de colores, resolución, cantidad de luz, enfoque etc. Por otro lado, encontramos áreas de mejora en modificar la cantidad de épocas en el reentrenamiento a 30,000 para reducir el costo computacional siendo que el comportamiento lineal se muestra a partir de 20,000 épocas.

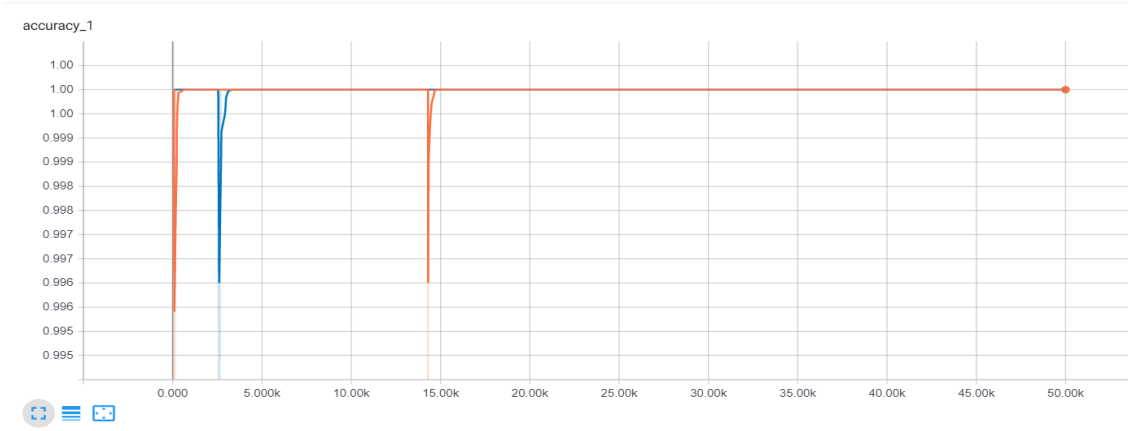


Figura 5.1 Gráfica de exactitud vs iteraciones.

En la Figura 4.2 se graficó el comportamiento del error en el reentrenamiento de la arquitectura GoogleNet. En la gráfica se muestran 2 líneas, la línea de color naranja representa el comportamiento del entrenamiento y la azul representa la validación. Los valores del eje X con la cantidad de épocas con las cuales se entrenó la arquitectura y los valores del eje Y representa el error estimado para clasificar imágenes de infantes.

Se puede observar que el comportamiento de las dos líneas en la gráfica no presenta una diferencia significativa. Lo que quiere decir que el modelo es robusto y capaz de clasificar los casos a estudiar. Al igual que en la figura 4.1 en la gráfica 4.2 se presenta una sección en la que las épocas tienen un mínimo cambio o variación en el error.

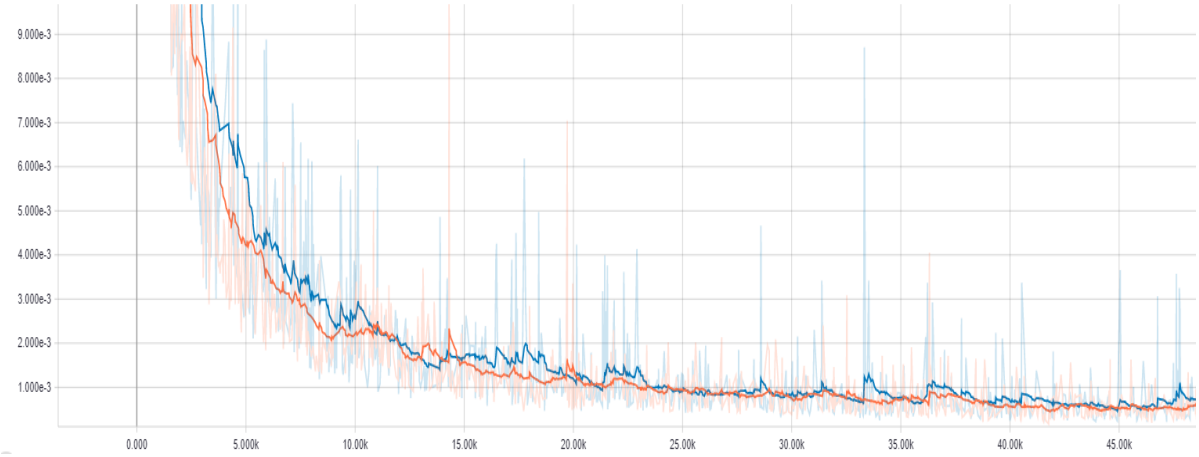


Figura 5.2 Gráfica de pérdida (Error).

Reforzando la teoría de reducir la cantidad de épocas en trabajos futuros en el reentrenamiento de la arquitectura GoogleNet. Se muestra en la figura 4.3 los valores estimados del error de entrenamiento y validación en los puntos de 30,000 y 50,000 épocas. En el entrenamiento un valor estimado en el error de 7.8730e-4 en 30,000 épocas y 2.6059e-4 en las 50,000 épocas, la variación que existe entre los dos vales es de 5.2671e-4. Por otra parte, se presenta los valores de validación en las 30,000 épocas de 4.6309e-4 y 4.0909e-4 en las 50,000 épocas. la variación estimada entre los dos puntos de corte es de 1.54e-4. Obteniendo variaciones pequeñas en el train y validación respecto a la gran cantidad de épocas y trabajo computacional que representa.

Name	Smoothed	Value	Step
train	8.0477e-4	7.8730e-4	29.83k
validation	8.2230e-4	4.6309e-4	29.78k

Name	Smoothed	Value	Step
train	5.4630e-4	2.6059e-4	50.00k
validation	7.6182e-4	3.0909e-4	50.00k

Figura 5.3 Error en 30K y 50K épocas.

Capítulo 6. Conclusiones

Se concluye que el modelo generado en la presente investigación es capaz de detectar a un infante en el asiento del copiloto, al obtener una precisión de 89.25% y una tasa de error de solo 10.75% durante la etapa de entrenamiento. Sin embargo, en la etapa de prueba a ciegas, la precisión aumentó a 97.66% con una tasa de error del 2.33%, lo cual demuestra la efectividad del modelo para detectar a un infante a través del análisis de imágenes capturadas de la zona del copiloto.

Asimismo, se confirma la efectividad de clasificación que tiene la arquitectura GoogleNet como red neuronal convolucional, ya que se aplicó sobre el conjunto de imágenes procesadas compuesto por un total de 59,729 imágenes, de las cuales 18,864 imágenes pertenecen a Infantes, 17,941 imágenes de Personas y 22,924 imágenes de diversos Objetos.

Cabe destacar que las imágenes procesadas son resultado del procesamiento detallado, de los videos grabados durante cada una de las pruebas realizadas para los participantes Infantes y Adultos y Objetos, durante el cual, cada imagen fue recopilada mediante el uso de una herramienta para la manipulación de videos.

Finalmente, en la presente se concluye la factibilidad que tiene la implementación de las técnicas de visión computacional y redes neuronales convolucionales para el desarrollo de un sistema de detección de infantes en la zona de copiloto, como una opción viable para la prevención y disminución de accidentes leves a fatales en infantes.

Así mismo, se demostró la efectividad de las técnicas de re-entrenamiento para adaptar una red con alto rendimiento a un problema nuevo, en este caso se demostró que la red pre-entrenada de google net, puede ser usada para ser usada con el propósito de detectar infantes en el asiento del copiloto.

La metodología propuesta tiene el potencial de reducir el número de accidentes en infantes una vez implementada a gran escala, mediante la prevención y alerta al conductor cuando éste expone a un infante al posicionarlo en el asiento del copiloto

Trabajo a futuro

Como trabajo a futuro se propone aumentar la cantidad de imágenes en las clases de Infantes y Objetos con el objetivo de balancear la cantidad de imágenes entre clases y evitar un sesgo o sobre ajuste entre clases.

Se propone implementar el modelo de detección sobre teléfonos móvil, en donde la detección de infantes se realice en tiempo real, con un prototipo viable, accesible y no invasivo para automóviles ordinarios.

Por otra parte, se propone aplicar el conjunto de imágenes generado sobre diversas arquitecturas de redes neuronales convolucionales pre-entrenadas como AlexNet, Inception V3, ResNet, entre otras, y determinar cuál de todas tiene el mayor valor de precisión.

Referencias

[1] M. Silva Martínez, “Educación vial,” *Bol. la Of. Sanit. Panam.*, vol. 77, no. 1, pp. 31–39, 1974.

[2] V. Noe, A. Garcia, and N. De, “RFC receptor : Nombre receptor : Folio fiscal : Conceptos Moneda :,” p. 970701, 2019.

[3] “Accidentes de tráfico en 1900 - Los primeros accidentes de tráfico,” 2015. [Online]. Available: https://vaiu.es/accidentes-de-trafico-en-1900/. [Accessed: 28-Nov-2019].

[4] “Así fue el primer accidente automovilístico de la historia | Turbo,” 2018. [Online]. Available: http://www.revistaturbo.com/noticias/asi-fue-el-primer-accidente-automovilistico-de-la-historia-887#. [Accessed: 28-Nov-2019].

[5] “OMS | 10 datos sobre la seguridad vial en el mundo,” *WHO*, 2017.

[6] “OPS/OMS México - OPS/OMS México.” [Online]. Available: https://www.paho.org/mex/index.php?option=com_content&view=article&id=552:mexico-ocupa-septimo-lugar-nivel-mundial-muertes-accidentes-transito-ops&Itemid=0. [Accessed: 09-Mar-2020].

[7] “Zacatecas, primer estado con mayor tasa de muertes por accidentes automovilísticos | Portada, Principales, Sociedad y Justicia | La Jornada Zacatecas.” [Online]. Available: https://ljz.mx/2019/11/01/zacatecas-primer-estado-con-mayor-tasa-de-muertes-por-accidentes-automovilisticos/. [Accessed: 20-Oct-2020].

[8] M. G. Fernández, “Primeros accidentes de la historia - Quo,” 2015. [Online]. Available: https://www.quo.es/tecnologia/g22908/primeros-accidentes-de-la-historia/. [Accessed: 28-Nov-2019].

[9] “La evolución de la seguridad en el automóvil - Autofácil,” 2014. [Online]. Available: https://www.autofacil.es/seguridad/2014/06/05/evolucion-seguridad-automovil/19056.html. [Accessed: 28-Nov-2019].

[10] J. M. Benítez, J. L. Castro, and I. Requena, “Are artificial neural networks black boxes?,” *IEEE Trans. Neural Networks*, vol. 8, no. 5, pp. 1156–1164, 1997.

[11] “Breve Historia de las Redes Neuronales Artificiales | Aprende Machine Learning,” 2018. [Online]. Available: https://www.aprendemachinelearning.com/breve-historia-de-las-redes-neuronales-artificiales/. [Accessed: 07-Jan-2020].

[12] D. E. Prevenci and D. E. L. A. Salud, *Programa de Acción : Accidentes Programa de Acción Accidentes*. .

[13] “México, séptimo lugar mundial en siniestros viales.” [Online]. Available: https://www.insp.mx/avisos/4761-seguridad-vial-accidentes-transito.html. [Accessed: 20-Oct-2020].

[14] Policía Federal, “Guía para prevenir accidentes de tránsito en jóvenes | Policía

Federal | Gobierno | gob.mx," 2018. [Online]. Available: https://www.gob.mx/policiafederal/articulos/guia-para-prevenir-accidentes-de-transito-en-jovenes?idiom=es. [Accessed: 28-Nov-2019].

[15] "Sistemas de retención infantil." [Online]. Available: https://www.seguridad-vial.net/educacion-vial/seguridad-infantil/55-sistemas-retencion-infantil. [Accessed: 28-Nov-2019].

[16] I. Kecerdasan and P. Ikep, "REQUEST FOR WAIVER OF SECTION," vol. 255, no. c, p. 6.

[17] "SensorSafe." [Online]. Available: https://sensorsafeapp.com/device-compatibility-eu.html. [Accessed: 19-Jan-2021].

[18] "OMS | Cada día mueren más de 2000 niños por lesiones no intencionales," *WHO*, 2013.

[19] secretaria de salud, "Legislación y Seguridad Vial," 2014. [Online]. Available: http://conapra.salud.gob.mx/Interior/Seguridad_vial_legislacion.html. [Accessed: 28-Nov-2019].

[20] "El uso de la inteligencia artificial en la prevención de accidentes laborales | Geseme." [Online]. Available: https://geseme.com/el-uso-de-la-inteligencia-artificial-en-la-prevencion-de-accidentes-laborales/. [Accessed: 16-Nov-2020].

[21] D. Torres, "Procesamiento digital de imágenes," *Perfiles Educ.*, no. 72, 1996.

[22] C. A. De and P. R. Lazo, "Procesamiento Digital de Imágenes Procesamiento Digital de Imágenes Procesamiento Digital de Imágenes Procesamiento Digital de Imágenes," *Ecologia*, pp. 1–4, 2006.

[23] D. D. E. Imágenes, D. Kantor, and M. Pellegrini, "Digitalización de imágenes," 2016.

[24] M. Ii, "Etapas de la visión artificial," 2008.

[25] Gonzales, "Digital image proccesing using MATLAB," *Igarss 2014*, no. 1, pp. 1–5, 2014.

[26] K. Pizer, S. M., Amburn, E. P., Austin, J. D., Cromartie, R., Geselowitz, A., Greer, T., ... & Zuiderveld, "Adaptive histogram equalization and its variations," *Computer Vision, Graphics, and Image Processing*, vol. 38, no. 1. p. 99, 1987.

[27] C. Peñafiel and R. Ávila, "Inteligencia Artificial," *Intel. Artif.*, vol. 2, no. 6, pp. 1–33, 2007.

[28] O. Simeone, "A Very Brief Introduction to Machine Learning with Applications to Communication Systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 648–664, 2018.

[29] D. J. Matich, "Redes Neuronales: Conceptos Básicos y Aplicaciones.," *Historia Santiago.*, p. 55, 2001.

[30] A. J.Serrano, “Redes Neuronales.”

[31] P. Larrañaga, “Tema 8. Redes Neuronales,” pp. 1–19.

[32] T. D. E. Grado and E. N. Ingenier, “Entrenamiento de redes neuronales basado en algoritmos evolutivos,” 2005.

[33] “¿Qué es una red neuronal?” [Online]. Available: https://www.ibm.com/support/knowledgecenter/es/SSLVMB_sub/statistics_mainhelp_ddita/spss/neural_network/nnet_whatis.html. [Accessed: 22-Oct-2020].

[34] J. Silva and J. Dorado, “Desarrollo De Redes De Neuronas Profundas Para Procesado De Imagen,” p. 90, 2019.

[35] L. Deng and D. Yu, “Deep learning: Methods and applications,” *Found. Trends Signal Process.*, vol. 7, no. 3–4, pp. 197–387, 2013.

[36] “Intro a las redes neuronales convolucionales | by Bootcamp AI | Medium.” [Online]. Available: https://bootcampai.medium.com/redes-neuronales-convolucionales-5e0ce960caf8. [Accessed: 15-Jan-2021].

[37] C. Szegedy *et al.*, “Going Deeper with Convolutions Christian,” *J. Chem. Technol. Biotechnol.*, vol. 91, no. 8, pp. 2322–2330, 2015.

[38] V. T. T. Mariano, “Introducción a las Redes Neuronales.” [Online]. Available: https://www.uaeh.edu.mx/docencia/P_Presentaciones/huejutla/sistemas/redes_neuronales/introduccion.pdf. [Accessed: 21-Oct-2020].

[39] P. Choudhary and N. R. Velaga, “En la transferencia de conocimiento no supervisado. La tarea de destino es diferente, pero relacionada con la tarea fuente. Sin embargo, la transferencia de conocimiento no supervisada se centra en la solución de tareas de aprendizaje no supervisado en el,” pp. 1–9, 2017.

[40] “Machine learning: conoce qué es y las diferencias entre sus tipos.” [Online]. Available: https://empresas.blogthinkbig.com/que-algoritmo-elegir-en-ml-aprendizaje/. [Accessed: 18-Jan-2021].

[41] V. Christanti Mawardi, N. Susanto, and D. Santun Naga, “Spelling Correction for Text Documents in Bahasa Indonesia Using Finite State Automata and Levinshtein Distance Method,” *MATEC Web Conf.*, vol. 164, pp. 1–15, 2018.

[42] J. Gonçalves, “Vehicle’s Interior Presence Detection and Notification System,” 2018.

[43] T. Mousel, P. Larsen, H. Lorenz, and I. S. A. Luxembourg, “Unattended Children in Cars-Radiofrequency-Based Detection To Reduce Heat Stroke Fatalities,” *25th Int. Tech. Conf. Enhanc. Saf. Veh.*, pp. 1–5, 2017.

[44] I. Applications, “Machine Learning-Based Human Recognition Scheme,” 2020.

[45] P. R. R. Devarakota, B. Mirbach, M. Castillo-Franco, and B. Ottersten, “3D vision technology for occupant detection and classification,” *Proc. Int. Conf. 3-*

D Digit. Imaging Model. 3DIM, pp. 72–79, 2005.

[46] K. Kasten, A. Stratmann, M. Munz, K. Dirscherl, and S. Lamers, “iBolt technology - A weight sensing system for advanced passenger safety,” *Adv. Microsystems Automot. Appl. 2006*, pp. 171–186, 2006.

[47] L. Federspiel, M. Cuddihy, and S. Fuks, “Signs By Seat-Embedded Ferroelectric Film Sensors and By Vibration,” pp. 1–8.

[48] M. E. Farmer and A. K. Jain, “Occupant classification system for automotive airbag suppression,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, 2003.

[49] J. J. Yoon, C. Koch, and T. J. Ellis, “Vision based occupant detection system by monocular 3D Surface Reconstruction,” *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, pp. 435–440, 2004.

[50] F. Bellard, “FFmpeg,” 2000. [Online]. Available: https://www.ffmpeg.org/. [Accessed: 19-Feb-2020].

[51] “OpenCV,” 1999. [Online]. Available: https://opencv.org/about/. [Accessed: 19-Feb-2020].

[52] A. Tharwat, “Classification assessment methods,” *Appl. Comput. Informatics*, Aug. 2018.

[53] “Bienvenido a Python.org.” [Online]. Available: https://www.python.org/. [Accessed: 30-Apr-2020].

[54] “R: El proyecto R para computación estadística.” [Online]. Available: https://www.r-project.org/. [Accessed: 30-Apr-2020].

[55] “RPubs - Introducción al paquete Caret.” [Online]. Available: https://rpubs.com/joser/caret. [Accessed: 30-Apr-2020].

[56] “NIR-Quimiometria: Confusion Matrix with Caret.” [Online]. Available: http://nir-quimiometria.blogspot.com/2018/10/confusion-matrix-with-caret.html. [Accessed: 30-Apr-2020].

[57] W. C. Almaraz, H. Luna-garcía, and J. M. Celaya-padilla, “Diseño de Feedback basado en DCU y primeras pruebas en Detección de Infantes con PDI de bajo nivel,” pp. 1–13, 1801.

[58] W. C. Almaraz *et al.*, “Children detection on passenger seat using UCD and IDP: An initial prototype,” *Commun. Comput. Inf. Sci.*, vol. 1114 CCIS, pp. 138–149, 2019.

Aportaciones Generales

Publicaciones de Investigación

Diseño de Feedback basado en DCU y primeras pruebas en Detección de Infantes con PDI de bajo nivel.

Autor principal del artículo publicado y presentado en las “V jornadas iberoamericanas de IHC” en la Benemérita Universidad Autónoma de Puebla, México.

El articulo propone un prototipo de alerta visual-auditiva para la prevención y disminución de lesiones graves en infantes. Mediante la metodología de Diseño centrado en el Usuario(UCD) [57].

Children Detection on Passenger Seat using UCD and IDP: An Initial Prototype.

Autor principal del artículo y publicado en el libro “Iberoamerican Workshop on Human-Computer Interaction” (2019).

El articulo presenta el desarrollo de un prototipo inicial para la detección de infantes en el asiento de copiloto mediante técnicas de procesamiento digital de imágenes de bajo nivel con alerta visual-auditiva para la prevención de lesiones y fallecimientos de infantes en accidentes automovilísticos [58].

Biolitex S.A de S.V.

Se consolido legalmente la empresa Biolitex, como resultado de la participación den la convocatoria INCUBATICZ 2019 – COZCYT.

La empresa Biolitex fue consolidada con el objetivo de crear biomarcadores usando bases de datos de población mexicana para una detección temprana, seguimiento

Anexos

Carta de consentimiento para grabar menores

Carta de Consentimiento

La principal causa de muerte en menores de 5 años está relacionada a accidentes dentro de ellos automovilísticos, debido a esto en la Universidad Autónoma de Zacatecas (UAZ) se está realizando una investigación para prevenir dichos accidentes, como parte de dicha investigación se está proponiendo un sistema que consta de video cámaras montadas al interior del vehículo el cual es capaz de detectar a un infante que se encuentra en alguna posición de riesgo, para que estos sistema puedan funcionar correctamente es necesario realizar pruebas experimentales por lo que se le invita a usted y a su niño de este experimento.

La prueba consistirá en un vehículo que se mantendrá **INMÓVIL** y se grabará una sesión de video en donde el niño(a) se sentará en asiento de copiloto (dependiendo de la edad del menor se usará un porta bebe), la duración máxima de la grabación será 5min.

Nota: El menor en ningún momento no correrá algún riesgo.

Consentimiento para la video grabación

yo: _____ con IFE/INE: _____ como Padre/Madre/Tutor legal de: _____

Doy consentimiento para:

hacer uso del material fotográfico y audiovisual que podrán ser utilizadas para:

- ➢ Participar en la video grabación arriba mencionado
- ☐ La cinta será utilizada única y exclusivamente para el análisis y desarrollo del proyecto de tesis" Detección de infantes como copilotos" dentro de la Universidad Autónoma de Zacatecas (UAZ).
- ☐ Publicarlas y presentarlas en trabajos de investigación educativa por el centro de investigación e innovación automotriz de México(CIIAM) y/o profesores de la unidad.
- ☐ Publicarlas en diferentes medios de comunicación (Artículos Científicos, Revistas, etc.) por parte de los organismos oficiales que realizan una actividad concreta para

dar constancia de lo que se ha hecho en la universidad, en cumplimiento de las leyes de protección de datos la identidad de los menores será mantenida anónima (La cara será distorsionada para proteger su identidad, no se usará su nombre, etc.)

_____

Firma

(nombre y firma)

Fecha___ /_____ /_____

Número del participante: _____

Hora: _____

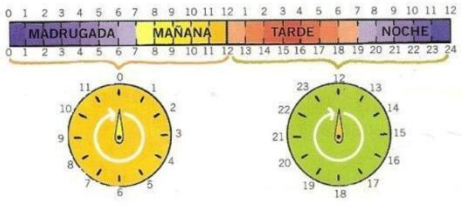
Ciclo del día: _____

Edad: _____

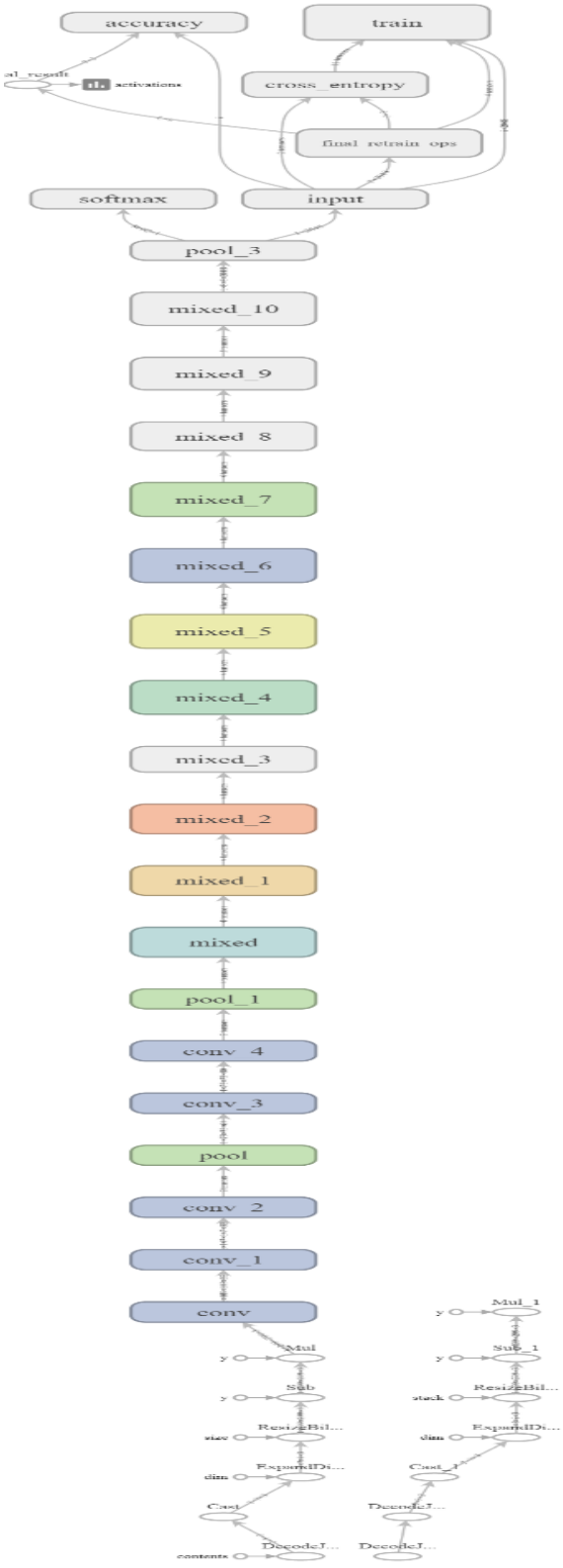
Estatura: _____

Peso: _____

Sexo: _____



Grafo final


accuracy
train
activations
cross_entropy
final_retrain_ops
softmax
input
pool_3
mixed_10
mixed_9
mixed_8
mixed_7
mixed_6
mixed_5
mixed_4
mixed_3
mixed_2
mixed_1
mixed
pool_1
conv_4
conv_3
pool
conv_2
conv_1
conv
Mul
Sub
ResizeBil...
ExpandDi...
Cast
DecodeJ...
contents
Mul_1
Sub_1
ResizeBil...
ExpandDi...
Cast_1
DecodeJ...
DecodeJ...