

UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

DIVISIÓN DE ESTUDIOS DE POSGRADO

“Sistema de asistencia a la conducción
implementando una red neuronal en un dispositivo
móvil”

Tesis

Para obtener el grado de:

Maestro en Robótica

Presenta:

Ing. Eduardo Aragón López

Director de tesis:

Dr. José Anibal Arias Aguilar

H. Cd. de Huajuapán de León, Oaxaca, México, Octubre del 2023

Sinodales:

Dr. Eduardo Sánchez Soto

Dr. Rosebet Miranda Luna

Dr. Oscar David Ramírez Cárdenas

Dr. Ignacio Arroyo Fernández

Director de Tesis:

Dr. José Anibal Arias Aguilar

Dedicatoria

Con especial cariño para mis padres y hermana quienes siempre han confiado en mí y me han animado en la continuación de mis estudios y en cada momento de mi vida.

Agradecimientos

Principalmente a Dios por guiarme en cada etapa de mi vida.

A mis padres Cuauhtémoc y Sabina por ser siempre mis mejores amigos, por brindarme su apoyo y su cariño, su amor ha sido mi principal motivación.

A mi hermana Soledad por su apoyo incondicional, su amistad y por estar conmigo en los momentos difíciles tratando de hacerme sonreír.

A mis abuelitos Lidio, Silvia, Genaro y Petronila quienes me han ayudado a ser la persona que soy ahora, especialmente a mi abuelito Lidio de quien adquirí el maravilloso sueño de los estudios y la docencia.

A toda mi familia quienes siempre han creído en mí, por todo el apoyo que me han brindado y por las reuniones familiares que siempre terminan con una sonrisa.

A Eduardo Ireta, un gran amigo quién me apoyó y motivó para ingresar a la maestría.

A mis amigos del IPN Edwin, Jero, José, Paula, Aimé con quienes a pesar del tiempo y la distancia continuamos teniendo una bonita amistad.

A mis amigos y profesores de la UTVCO quienes siguen apoyándome y motivándome a seguir avanzando y a llegar más lejos.

A mis amigos de la UTM Cynthia Mendoza, Aby, Alan, Kevin, Rosario, Fernando, Jesús, Ana Paola, Canseco, Frida, Citlally, Deisy, Karla, Ross y Denni por haberme dado la bienvenida a una nueva etapa de mi vida, por ayudarme a llevar los días complicados con optimismo, por creer en mí y los momentos de diversión que

compartimos. En especial agradezco la amistad de Diana Laura, con quien adquirí una fuerte amistad, por su apoyo incondicional y sus consejos, gracias por estar conmigo siempre.

Al club de lectura por ayudarme a salir de la rutina de los estudios, especialmente agradezco a Cynthia Torrevela y Rubí, quienes me ayudaban a integrarme más al grupo.

A mis amigos Juan Carlos, Xare, Pam, Wendy, Pedro, Emmanuel, Omar Alí, Milo, Venegas, Alexis y Lúa por llenarme de alegría cada fin de semana, por su música, sus consejos y sus motivaciones, estoy agradecido con la vida por haberlos conocido.

Al Dr. José Anibal Arias Aguilar por sus enseñanzas y apoyo como director de tesis, por encaminarme en el desarrollo de este proyecto y solventar mis dudas.

A mis sinodales, profesores y personal de la UTM por su apoyo durante la maestría.

Un agradecimiento a todas mis amistades y familiares que han formado parte de mi vida y que no alcancé a nombrar en estos agradecimientos.

Y finalmente agradezco al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCyT) por la beca otorgada (Becario No. CVU: 1151734).

RESUMEN

El presente trabajo de tesis aborda el desarrollo y la implementación en un dispositivo móvil de un Sistema de Asistencia a la Conducción, que genera recomendaciones sonoras a los conductores de automóviles. Mediante el entrenamiento de algoritmos de aprendizaje profundo basados en redes neuronales convolucionales, este proyecto aportó un sistema de reconocimiento de distracciones comunes en conductores implementado en una aplicación Android para la alerta temprana de la conducción distraída. Con base en los datos capturados por el INEGI (2020) se puede observar que el automóvil es el tipo de vehículo que mayormente sufre accidentes viales, por tal motivo el presente trabajo se enfoca en este tipo de transporte.

Mediante el análisis de bases de datos y la generación de una base de datos local de conductores de automóviles se realizaron entrenamientos con diversos modelos de redes neuronales convolucionales, utilizando la técnica de “transferencia de aprendizaje” se obtuvo un mejor modelo capaz de detectar la conducción segura y nueve distracciones en conductores de automóviles: Enviando SMS con mano derecha, Llamada con mano derecha, Enviando SMS con mano izquierda, Llamada con mano izquierda, Manipulando la radio/GPS, Ingiriendo bebidas, Buscando algo en la parte de atrás, Maquillando/Arreglando el cabello, Hablando con pasajero. Mismo modelo se implementó en un dispositivo móvil por medio de una aplicación Android.

ÍNDICE

RESUMEN.....	ix
ÍNDICE	xi
LISTA DE FIGURAS	xv
LISTA DE TABLAS.....	xvii
Capítulo I: INTRODUCCIÓN	1
Antecedentes	3
1.1 Antecedentes comerciales.....	3
1.1.1 Attention assist.....	3
1.1.2 Smart eye.....	4
1.1.3 Driver Attention Monitoring.....	5
1.1.4 Tecnología ADAS Peugeot.	6
1.2 Antecedentes de investigación.....	7
1.2.1 Asistente de conducción para la detección de somnolencia.....	8
1.2.2 Red neuronal para reconocer distracciones en la conducción.....	9
1.2.3 Monitorización del conductor en Android	10
1.2.4 Detección de emociones	11
1.2.5 Sistema de detección de fatiga en conductores.....	12
1.2.6 Reconocimiento de emociones en conductores	13
Capítulo II: Anteproyecto.....	15
2.1 PLANTEAMIENTO DEL PROBLEMA	15
2.1.1 Motivación	16
2.2 JUSTIFICACIÓN.....	19
2.2.1 Pertinencia	19
2.2.2 Relevancia	19
2.3 HIPÓTESIS.....	20
2.4 OBJETIVOS.....	20
2.4.1 Objetivo general.....	20
2.4.2 Objetivos específicos	20
2.5 METAS	21
2.6 LIMITACIONES DE LA TESIS.....	22

2.7 ORGANIZACIÓN DE LA TESIS	22
Capítulo III: Marco teórico	25
3.1 Visión artificial	25
3.1.1 Dificultades de la visión artificial	27
3.1.2 Variaciones en la iluminación	28
3.1.3 Movimiento	29
3.2 Imagen digitalizada	29
3.3 Procesamiento digital de imágenes	30
3.3.1 Conversión de imágenes de 3 canales RGB a 2 canales escala de grises	31
3.4 Inteligencia artificial	32
3.4.1 Detección de objetos en imágenes	34
3.4.2 Computación afectiva	35
3.5 Aprendizaje automático	36
3.5.1 Aprendizaje supervisado	37
3.5.2 Aprendizaje no supervisado	39
3.5.3 Aprendizaje por refuerzo	40
3.6 Redes neuronales artificiales	41
3.6.1 Redes artificiales monocapa	42
3.6.2 Redes neuronales artificiales multicapa	43
3.6.3 Redes neuronales artificiales profundas	45
3.7 Redes neuronales convolucionales	46
3.7.1 Kernel	47
3.7.2 Etapa convolucional	47
3.7.3 Capa de pooling	48
3.7.4 Aplanado de los datos	49
3.7.5 Etapa de clasificación	50
3.7.6 Tasa de aprendizaje	50
3.7.7 Tamaño del lote	51
3.7.8 Número de épocas	51
3.8 Transferencia de aprendizaje	52
3.9 Sistema ADAS	53
Capítulo IV: Metodología y desarrollo	55
4.1 Etapa 1: Detección de distracciones en los conductores	57

4.2 Etapa 2: Definición de distracciones para el proyecto	57
4.3 Etapa 3: Análisis de las bases de datos	61
4.3.1 Protocolo de realización de pruebas con base de datos “State Farm Distracted Driver Detection”	70
4.3.2 Preprocesamiento de las imágenes	72
4.3.3 Desarrollo de los modelos y entrenamientos desde cero	74
4.3.4 Transferencia de aprendizaje	83
4.3.5 Modelo óptimo entrenado desde cero	86
4.3.6 Evaluación del modelo con mejores resultados	89
4.4 Etapa 4: Protocolo de captura de imágenes y generación de la base de datos local	93
4.4.1 Paso 1: Selección de participantes	95
4.4.2 Paso 2: Captura de las imágenes	96
4.4.3 Paso 3: Etiquetado de las imágenes	97
4.4.4 Paso 4: Generación de la base de datos	98
4.5 Etapa 5: Protocolo de entrenamiento con transferencia de aprendizaje a la base de datos generada	99
4.5.1 Paso 1: Preprocesamiento de las imágenes	101
4.5.2 Paso 2: Selección de modelos preentrenados	102
4.5.3 Paso 3: Ajuste de hiperparámetros y entrenamiento aplicando transferencia de aprendizaje a la base de datos generada	106
4.5.4 Paso 4: Evaluación del modelo con mejores resultados	109
4.6 Etapa 6: Implementación móvil	110
4.7 Etapa 7: Aplicación en funcionamiento	116
Capítulo V: Resultados y discusión	119
Capítulo VI: Conclusiones.....	127
6.1 Trabajo a futuro	127
REFERENCIAS	129
Apéndice A	135
Publicaciones	135
Apéndice B	137
Convocatorias	137

LISTA DE FIGURAS

Ilustración 1 ATTENTION ASSIST obtenido de (Mercedes-Benz , 2019).	4
Ilustración 2 Tecnología de seguridad para interiores de automóviles obtenido de (Smart Eye , 2021)	5
Ilustración 3 Detección y monitoreo del conductor obtenido de (CARMANÍA, 2020).	6
Ilustración 4 ADAS Peugeot obtenido de (Peugeot, 2020).	7
Ilustración 5 Sistema de visión con iluminación infrarroja obtenido de (Calero, Moreno, & Hueso, 2010).	9
Ilustración 6 Cara de perfil sin detección obtenido de (Salcedo, 2015).	10
Ilustración 7 Modelo de la red neuronal convolucional utilizada obtenido de (Jaramillo Romero, 2020).	11
Ilustración 8 Ejemplo de detección de bostezo y parpadeo obtenido de (Paterna, 2020).	12
Ilustración 9 Ejecución del sistema obtenido de (Flórez, 2020).	13
Ilustración 10 Accidentes de tránsito terrestre, vehículos involucrados obtenido de (INEGI 2020).	17
Ilustración 11 Ejemplo de la captura de una imagen y su procesamiento digital obtenido de Azuela, (2013).	26
Ilustración 12 Ejemplo de clasificación de dos clases obtenido de (Fabio & Manuela, 2012).	27
Ilustración 13 Clases parecidas entre sí	28
Ilustración 14 Variaciones en la iluminación obtenido de (Goro & Ashish, 2007).	28
Ilustración 15 Desenfoque de la imagen por movimiento obtenido de (Goro & Ashish, 2007).	29
Ilustración 16 El espacio de coordenadas cartesianas 2-D de una imagen digital M x N obtenido de (Murguía, 2007).	30
Ilustración 17 Mejora y extracción de características de la imagen obtenido de (Murguía, 2007).	31
Ilustración 18 Conversión de imagen a color a escala de grises obtenido de (Roberto & Juan, 2012). 32	
Ilustración 19 Características de la IA.	33
Ilustración 20 Modelo funcional del reconocimiento de imágenes obtenido de (Yildirim, 2010).	35
Ilustración 21 Subconjuntos del aprendizaje automático.	37
Ilustración 22 Modelo de aprendizaje supervisado obtenido de (González, 2018).	38
Ilustración 23 Aprendizaje no supervisado obtenido de (González, 2018).	39
Ilustración 24 Modelo de aprendizaje por refuerzo obtenido de (González, 2018).	40
Ilustración 25 Modelo estándar de una neurona artificial obtenido de (Berzal, 2019).	42
Ilustración 26 Modelo de red neuronal monocapa obtenido de (Michie, Spiegelhalter, & Taylor, 1994).	43
Ilustración 27 Modelo de una red neuronal multicapa obtenido de (Simon, 2009).	44
Ilustración 28 Clasificador lineal y no lineal.	45
Ilustración 29 Modelo de una red neuronal artificial profunda obtenido de (López & Fernández, 2008).	46
Ilustración 30 Ejemplo de la convolución aplicada a una imagen obtenido de (Berzal, 2019).	48
Ilustración 31 Aplanado de los datos.	49

Ilustración 32 Ejemplo de las etapas de una red neuronal convolucional obtenido de (Gomilla, 2019).	50
Ilustración 33 Metodología en cascada.	56
Ilustración 34 Tiempo y distancias recorridas por distracciones obtenido de (Dirección General de Tráfico de España, 2023).	58
Ilustración 35 Distancia de frenado ante imprevistos obtenido de (Instituto Mexicano del Transporte, 2022).	59
Ilustración 36 Diferentes perspectivas de la cámara	62
Ilustración 37 Imágenes de JAFFE	63
Ilustración 38 Imágenes de 3MDAD	64
Ilustración 39 Imágenes de FER	66
Ilustración 40 Imágenes de SFDDD	66
Ilustración 41 Imágenes de la base de datos	69
Ilustración 42 Protocolo de entrenamiento para los modelos	70
Ilustración 43 Ajuste de las imágenes	73
Ilustración 44 Arquitectura “MobileNet-v2” obtenido de (Sandler, Howard, Zhu, Zhmoginov y Chen, 2018).	84
Ilustración 45 Arquitectura “VGG-16” obtenido de (MATLAB, 2023).	85
Ilustración 46 Modelo óptimo	88
Ilustración 47 Gráfica de exactitud	90
Ilustración 48 Gráfica de pérdida	91
Ilustración 49 Matriz de confusión del modelo	92
Ilustración 50 Protocolo de captura de imágenes y generación de la base de datos	94
Ilustración 51 Soporte para celular	96
Ilustración 52 Etiquetado de la base de datos	98
Ilustración 53 Base de datos propia	99
Ilustración 54 Protocolo de entrenamiento de modelos preentrenados	100
Ilustración 55 Redimensionamiento de la base de datos local	101
Ilustración 56 Arquitectura de la red “Efficientnetb0” obtenido de (Keras, 2023).	103
Ilustración 57 Arquitectura de la red “Efficientnetb1” obtenido de (Keras, 2023).	104
Ilustración 58 Arquitectura de la red “Efficientnetb2” obtenido de (Keras, 2023).	105
Ilustración 59 Exactitud del modelo con la base de datos generada	109
Ilustración 60 Pérdida del modelo con la base de datos generada	110
Ilustración 61 Conversión del modelo a TensorFlow Lite	111
Ilustración 62 Conversión del modelo a formato TF Lite	112
Ilustración 63 Evaluación de la aplicación	113
Ilustración 64 Unión del modelo con los metadatos	114
Ilustración 65 Carga del modelo TF Lite en la aplicación	115
Ilustración 66 Recomendación del sistema	116
Ilustración 67 Aplicación en funcionamiento	117

LISTA DE TABLAS

Tabla 1 Metodología en cascada ventajas y desventajas.....	55
Tabla 2 Base de datos JAFFE.....	63
Tabla 3 Base de datos 3MDAD	64
Tabla 4 Base de datos FER.....	65
Tabla 5 Base de datos SFDDD.....	67
Tabla 6 Clases del Dataset.....	69
Tabla 7 Hiperparámetros de los primeros 4 modelos.....	76
Tabla 8 Arquitectura de la red del modelo 1	77
Tabla 9 Arquitectura de la red modelo 2	78
Tabla 10 Arquitectura de la red modelo 3	79
Tabla 11 Arquitectura de la red modelo 4	80
Tabla 12 Resultados de las pruebas de los modelos 1 al 4	81
Tabla 13 Variación de los hiperparámetros	81
Tabla 14 Resultados de las pruebas	82
Tabla 15 Respuesta de redes preentrenadas.....	86
Tabla 16 Nuevos hiperparámetros.....	87
Tabla 17 Resultados del modelo	89
Tabla 18 Representación de las etiquetas de la matriz de confusión.....	93
Tabla 19 Hiperparámetros para los primeros 4 modelos con la base de datos local	106
Tabla 20 Resultados de los primeros 4 entrenamientos.....	107
Tabla 21 Nuevos hiperparámetros para base de datos local.....	108
Tabla 22 Nuevos entrenamientos con nuevos hiperparámetros en la base local	108
Tabla 23 Comparación con antecedentes comerciales 1.....	121
Tabla 24 Comparación con antecedentes comerciales 2.....	122
Tabla 25 Comparación con antecedentes de investigación 1.....	123
Tabla 26 Comparación con antecedentes de investigación 2.....	124

Capítulo I: INTRODUCCIÓN

Conducir es una acción compleja que nos exige mucho más de lo que parece, involucra al conductor psicológica y físicamente (Isoba, 2021). El logro de una conducción correcta depende no sólo de las habilidades del conductor para dominar su vehículo en todo momento y circunstancia, sino que también depende de su estado emocional y de las distracciones que su entorno le pueda generar, lo cual puede llegar a causar una conducción distraída, son ejemplo de esta, la velocidad y el alcohol, el uso del teléfono móvil, la somnolencia, ingerir alimentos y bebidas, maquillarse y otros distractores más los causantes de alrededor del 90% de los siniestros viales (García, 2021).

Los sistemas de seguridad y asistentes de conducción en automóviles desempeñan una labor importante en la prevención de distracciones y accidentes viales (Volkswagen, 2020). Por esta razón, tanto la industria automotriz como la comunidad científica han estado en la constante búsqueda de poder mejorar la tecnología implementada en los asistentes a la conducción para poder incorporarlos en los automóviles y optimizar su funcionamiento, para ser capaces de detectar actividades potencialmente peligrosas para los conductores, las cuales pueden llegar a ocasionar accidentes viales, y que estos sean capaces de sugerir recomendaciones a los conductores, por ejemplo, de cuando deben de tomar un descanso, etc.

En años recientes estos sistemas de asistencia han ido teniendo mejoras y se les han ido incorporando sensores que detectan la presencia de las manos en el

volante; otros sistemas más sofisticados incluso son capaces de detectar y analizar el ritmo cardiaco del conductor para determinar si su estado físico es el idóneo para la conducción (Toyota, 2018). Otra incorporación que han tenido estos sistemas es el uso de cámaras en tiempo real, capaces de analizar al conductor e inclusive a su entorno, cuando estos sistemas detectan algún desvío de la atención del conductor fuera de lo establecido (e.g., desviar la mirada para utilizar la radio, hablar con el pasajero, maquillarse o peinarse, uso de dispositivos móviles) se alerta al conductor para que éste tome las medidas necesarias.

En el presente trabajo se implementó un Sistema de Asistencia a la Conducción por medio de un modelo de red neuronal entrenada, el cual mediante el uso de la “transferencia de aprendizaje” y una base de datos generada con 10,000 imágenes obtuvo un porcentaje de exactitud de alrededor del 90%, el cual detecta 10 clases, las cuales son, la conducción segura y nueve distracciones en conductores de automóviles: Enviando SMS con mano derecha, Llamada con mano derecha, Enviando SMS con mano izquierda, Llamada con mano izquierda, Manipulando la radio/GPS, Ingeriendo bebidas, Buscando algo en la parte de atrás, Maquillando/Arreglando el cabello, Hablando con pasajero. Dicho modelo fue procesado mediante la herramienta *TensorFlow Lite* para ser adaptado a una aplicación *Android* y de esta manera ser capaz de asistir a los conductores de automóviles, contribuyendo a la detección y alerta temprana de la conducción distraída y en la mejora de la tecnología utilizada en los Sistemas de Asistencia a la Conducción, lo cual a su vez puede ayudar en la disminución del índice de accidentes viales por distractores en los conductores.

Antecedentes

1.1 Antecedentes comerciales

Los Sistemas de Asistencia a la Conducción (*ADAS*) son tecnologías que buscan auxiliar al conductor en determinadas situaciones, con la finalidad de reducir la gravedad de los accidentes y evitar en lo posible las muertes que puedan provocar a causa de la falta de atención por parte del conductor, buscando de esta manera poder disminuir el número de accidentes automovilísticos. Los *ADAS* son sistemas inteligentes incorporados dentro del vehículo que ayudan al conductor de diversas maneras, mediante sensores o un monitoreo constante del conductor y del entorno pueden facilitar tareas como estacionarse, algunos permiten detectar autos en los puntos ciegos del automóvil e incluso detectar si el conductor está prestando su total atención en la conducción o si se está distraendo (Planing, 2014).

1.1.1 Attention assist.

Mercedes-Benz (2019) introdujo *ATTENTION ASSIST* (Ilustración 1) para ayudar a aumentar la seguridad en la conducción, especialmente en viajes largos o de noche y evitar el peligro de micro sueños. El sistema al identificar síntomas característicos de fatiga y distracción, muestra al conductor mediante alarmas visuales y acústicas para sugerirle que tome un tiempo de descanso.

Para determinar las acciones del conductor, el ADAS analiza el patrón de movimientos del volante y crea una base de datos con el perfil individual de la forma de conducir del conductor. Este perfil se compara con los datos de los sensores y con la base de datos que se tiene del perfil de conducción pudiendo detectar pequeños cambios en la conducción que puedan alertar de una fase temprana de la somnolencia y prevenir accidentes.



Ilustración 1 ATTENTION ASSIST obtenido de (Mercedes-Benz , 2019).

1.1.2 Smart eye.

La tecnología de seguimiento de ojos, boca y rostro apoyada en el uso de la inteligencia artificial de Smart Eye (2021) recaba información para analizar el estado del conductor, su concentración y somnolencia durante el camino. El propósito del ADAS es monitorear al conductor y asistirlo durante los indicios de cansancio y responder dependiendo de la situación.

Su función se centra en realizar monitorizaciones constantes del conductor para evaluar el número y frecuencia de parpadeos, la dirección y la orientación de la cara para la estimación de la somnolencia y de asegurarse de que su enfoque permanezca en la carretera, en la Ilustración 2 se aprecia un ejemplo del monitoreo.



Ilustración 2 Tecnología de seguridad para interiores de automóviles obtenido de (Smart Eye , 2021) .

1.1.3 Driver Attention Monitoring.

La tecnología de *DS Automobiles Driver Attention Monitoring* contribuye a conservar la atención del conductor lo que disminuye el riesgo de quedarse dormido al volante. El ADAS utiliza un par de cámaras infrarrojas para dar un seguimiento constante al conductor junto con la posición del vehículo. Las cámaras evalúan tres importantes signos de distracción y somnolencia: desvío de mirada, parpadeos y el

cuello (Ilustración 3). Cuando el sistema detecta alguno de los síntomas lanza una alarma audible y visual en el tablero digital del vehículo.

La implementación de estas tecnologías permite monitorear, minuciosamente al conductor buscando indicios de distracción y somnolencia analizando los movimientos que haga la cabeza del conductor, frecuencia de parpadeos, y desvío de la mirada. Además, cuenta con la facultad de incrementar el tiempo en el que se puede notificar al conductor para recomendarle que tome medidas preventivas contra los síntomas presentados (CARMANÍA, 2020).



Ilustración 3 Detección y monitoreo del conductor obtenido de (CARMANÍA, 2020).

1.1.4 Tecnología ADAS Peugeot.

La tecnología de *Peugeot* (Ilustración 4) ayuda en la prevención y detección de falta de atención, le ayudan al conductor para mantenerse alerta al conducir para una mayor seguridad durante el viaje en la carretera.

El sistema cuenta con un notificador de tiempo de conducción, el cual le recuerda al conductor tomar descansos frecuentes cuando se desplaza a través de largas distancias, de igual forma se apoya de una cámara que monitorea las líneas continuas de la carretera, de esta forma identifica las desviaciones relativas a las marcas viales para determinar si la conducta del automóvil en movimiento es sinónimo de síntomas de somnolencia o debido a una falta de atención de la persona.

El ADAS envía una alarma cuando se percata de que tras dos horas de conducción no se ha tomado ningún descanso, la alarma enviada se compone de una señal sonora cuya intención es recomendar un descanso al conductor y si este no toma en cuenta la recomendación, la alerta sonora continuará cada hora hasta que se realice el descanso recomendado (Peugeot, 2020).



Ilustración 4 ADAS Peugeot obtenido de (Peugeot, 2020).

1.2 Antecedentes de investigación

Actualmente en el aprendizaje automático las redes neuronales artificiales se han convertido en una poderosa herramienta. Sarmiento-Ramos (2020) comenta que tienen como objetivo desarrollar sistemas con la capacidad de aprender de forma automática a base de entrenamientos, reconociendo patrones, prediciendo comportamientos y generalizando información a partir de conjuntos de datos. Esta herramienta se ha convertido en un potencial campo de investigación con aplicaciones en los Sistemas de Asistencia a la Conducción y en muchas otras áreas como lo son la clasificación de imágenes, el reconocimiento de objetos y expresiones.

1.2.1 Asistente de conducción para la detección de somnolencia.

En el proyecto de Calero, Moreno, y Hueso (2010) de la Universidad Carlos III de Madrid, se desarrolló un ADAS, el cual mediante el monitoreo el conductor a través de una cámara es capaz de detectar principios de somnolencia del conductor y a su vez su posible distracción, lo cual constituye la metodología más apropiada por ser fiable, no invasiva y no intrusiva, es decir, el sistema se basa en un análisis constante del conductor por medio de una cámara en lugar de utilizar sensores que requieran adherirse al conductor para monitorearlo, tales como, el del ritmo cardíaco o algún otro sensor que requiera estar en contacto con el conductor y pueda provocarle una incomodidad o una distracción, motivo por el cual el sistema presentado no interfiere en la conducción.

El ADAS logra alcanzar un porcentaje de exactitud del 90% al enfocarse en el análisis de la cara y mirada del conductor en el transcurso del tiempo y elaborar un

índice de somnolencia y distracción. Los índices se generaron utilizando técnicas del aprendizaje automático (Máquinas de Soporte Vectorial) aplicadas en las imágenes tomadas mediante una cámara de visión infrarroja (Ilustración 5).



Ilustración 5 Sistema de visión con iluminación infrarroja obtenido de (Calero, Moreno, & Hueso, 2010).

1.2.2 Red neuronal para reconocer distracciones en la conducción.

En el trabajo de Núñez (2021) se implementó una red neuronal convolucional, la cual alcanza un porcentaje de exactitud del 85% para la detección de distracciones del conductor que puedan ocasionar accidentes de tránsito, a partir del reconocimiento de los estados de ánimo (feliz, enojo, tristeza, neutro, sorpresa) de los conductores. El sistema puede sugerir al usuario que tome descansos o sus debidas precauciones.

Este sistema se basa en analizar el comportamiento del conductor por medio de una cámara y se compara con una base de datos, la cual le ayudará al sistema a tomar una decisión sobre qué tipo de emoción es la que está afectando al usuario.

1.2.3 Monitorización del conductor en Android

En el proyecto de Salcedo (2015) de la Universidad Carlos III de Madrid se utiliza la cámara de un dispositivo móvil para obtener información del conductor, su finalidad es poder asistir en un viaje seguro al usuario, logrando una confianza del 75%.

La aplicación Android utiliza la cámara del dispositivo móvil para iniciar una búsqueda de rostros, y una vez detectado alguno comienza a realizar un monitoreo del conductor. Los rostros son detectados únicamente cuando la persona mira hacia el frente, por lo tanto, el sistema dejará de detectar el rostro cuando la persona mire hacia cualquier lado que no sea la posición frontal como se puede apreciar en la Ilustración 6. Por lo cual, la aplicación detectará la distracción del conductor cuando éste no tenga su atención puesta en la carretera y este desviando su atención en alguna otra actividad que no sea la conducción. Al no detectar algún rostro el ADAS da inicio a una cuenta regresiva y si en 2.5 segundos el ADAS sigue sin detectar al conductor, activa un mensaje de alarma para notificar la distracción detectada.

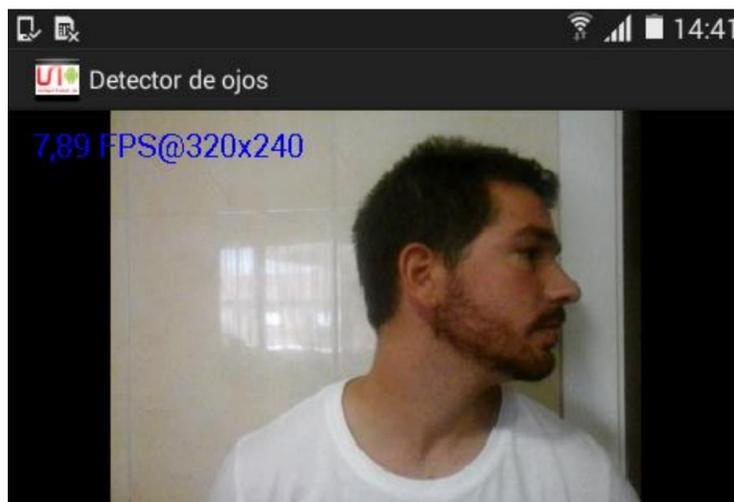


Ilustración 6 Cara de perfil sin detección obtenido de (Salcedo, 2015).

1.2.4 Detección de emociones

El proyecto de Jaramillo Romero (2020) de la Universidad Técnica de Machala consiste en el desarrollo de un sistema de reconocimiento facial y evaluación de emociones (Ilustración 7) mediante el uso de técnicas de aprendizaje profundo, como lo son las redes neuronales convolucionales. La red neuronal es capaz de realizar la detección de cinco clases (feliz, neutral, enojado, sorprendido y triste) y emplea el software de programación Python, en conjunto con la librería de visión artificial OpenCV, la cual es una librería de software de código abierto utilizada comúnmente en temas de aprendizaje automático y visión por computadora.

La red neuronal alcanzó una precisión del 67.13%, un buen porcentaje teniendo en cuenta la complejidad del problema y la ambigüedad de las clases, la base de datos cuenta con un total de 3000 imágenes por cada clase y tomando en cuenta que clasificar emociones en imágenes de rostros es una tarea difícil debido a que una misma imagen podría tener más de una categoría, lo que notablemente afecta a la exactitud de una red neuronal.

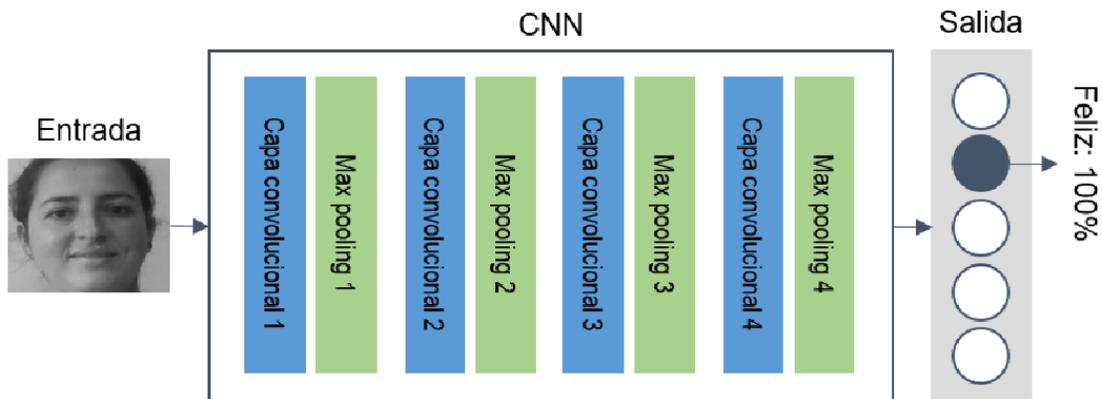


Ilustración 7 Modelo de la red neuronal convolucional utilizada obtenido de (Jaramillo Romero, 2020).

1.2.5 Sistema de detección de fatiga en conductores

En el proyecto de Paterna (2020) de la Universidad Politécnica de Cartagena se desarrolló un Sistema Avanzado de Asistencia a la Conducción (ADAS) dedicado a identificar la etapa temprana de fatiga en conductores con un porcentaje de exactitud del 80%. El ADAS se elaboró siguiendo un proceso dividido en dos etapas, la detección del rostro del conductor, y el análisis de la apertura de la boca y de los ojos en busca de síntomas de somnolencia. Los valores de apertura de boca y ojos fueron obtenidos haciendo uso de la librería OpenCV. La decisión se basa en el uso de un umbral ajustable el cual se encarga de identificar cuando el conductor se está quedando dormido al monitorear en tiempo real la apertura de los ojos y los bostezos. Cuando la apertura de la boca y ojos sobrepase el umbral dedicado a compararlos durante un cierto tiempo establecido, se lanzará un aviso indicando que el conductor se está quedando dormido (Ilustración 8).

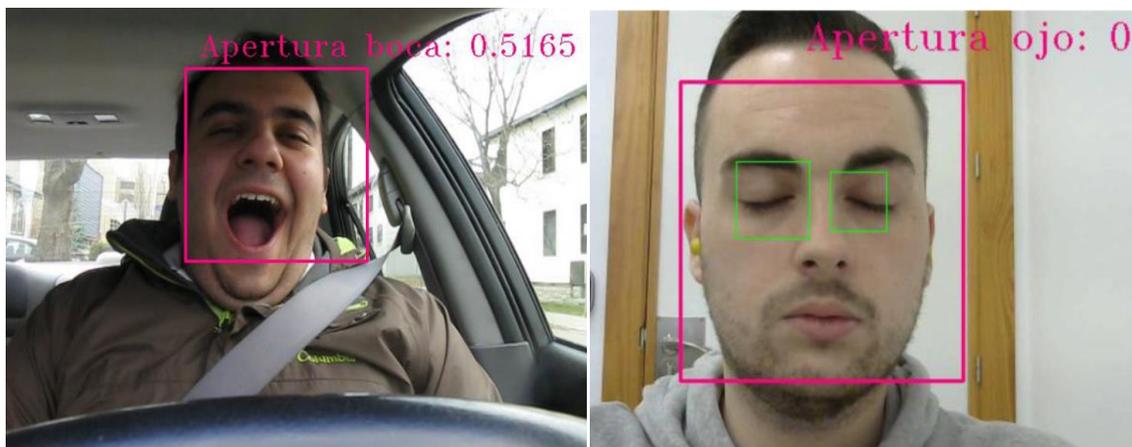


Ilustración 8 Ejemplo de detección de bostezo y parpadeo obtenido de (Paterna, 2020).

1.2.6 Reconocimiento de emociones en conductores

En el trabajo de Flórez (2020) de la Universidad Carlos III de Madrid se diseña un ADAS. Este sistema alcanza un porcentaje del 85% en exactitud, mediante la aplicación del aprendizaje automático para realizar un monitoreo del estado de ánimo y de las acciones que realiza el conductor (Ilustración 9).

El ADAS realiza una ejecución de dos modelos de redes neuronales convolucionales, cuya función es determinar el grado de distracción del conductor, haciendo uso de una cámara que estará capturando imágenes constantemente, ubicada dentro del vehículo. La primer red neuronal predice la emoción de la persona, mientras que la segunda red evalúa las acciones que realice el conductor. Cuando se presenta una acción o estado de ánimo peligroso se propone que el sistema tome una acción, caso contrario el sistema continúa monitorizando al conductor.

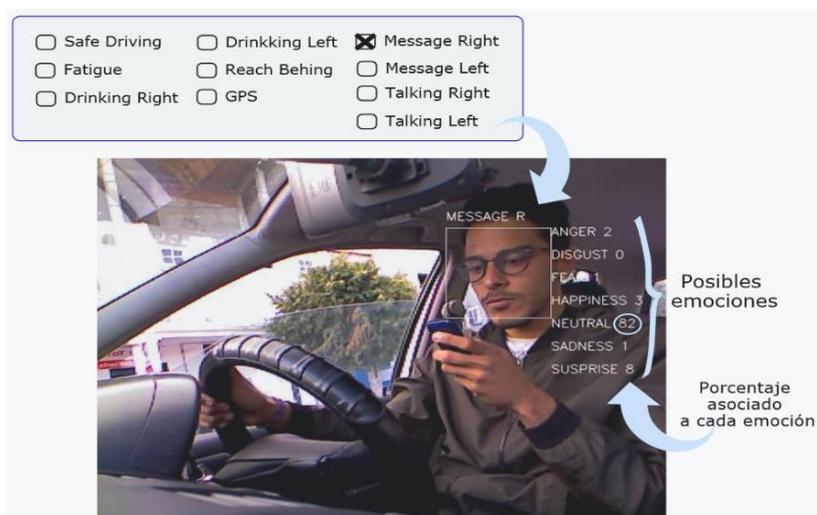


Ilustración 9 Ejecución del sistema obtenido de (Flórez, 2020).

Capítulo II: Anteproyecto

2.1 PLANTEAMIENTO DEL PROBLEMA

El presente proyecto de investigación aborda la problemática de la detección y alerta temprana de la conducción distraída aplicada a Sistemas de Asistencia a la Conducción y su integración con la clasificación de imágenes con redes neuronales convolucionales en una aplicación móvil. Como se mencionó en la sección de Antecedentes los ADAS han buscado ser un apoyo para los conductores al buscar formas de detectar signos de distracción en los mismos, ya que como mencionan Lipovac, Đerić, Tešić, Andrić y Marić (2017) la conducción distraída es de los principales factores que pueden ocasionar algún tipo de accidente vial, así mismo estos sistemas han implementado técnicas novedosas como el uso de cámaras que dan un seguimiento del rostro del conductor, sin embargo, en la mayoría de los trabajos presentados no se ha aplicado el uso de las redes neuronales convolucionales integrados en una aplicación móvil.

Bajo este marco de referencia se propone el siguiente proyecto capaz de realizar la detección y alerta temprana de la conducción distraída por medio de la identificación de la conducción segura y un conjunto de 9 distracciones en conductores de automóviles.

2.1.1 Motivación

De acuerdo a la información proporcionada por la Organización Mundial de la Salud (2019) acerca del alto índice y mortalidad provocados por los siniestros viales, se obtuvo que estos mismos se han convertido en el octavo principal factor de mortandad a una escala mundial. Estadísticas proporcionadas por el Instituto Nacional de Salud Pública (2020) demuestran que anualmente en el mundo se registran decesos próximos a los 1,3 millones de personas que son víctimas de los percances viales. Mismo informe reveló que “México ocupa el séptimo lugar a nivel mundial y el tercero en la región de Latinoamérica en muertes por siniestros viales”; los incidentes en materia vial alcanzaron posicionarse dentro de los diez primeros motivos de decesos durante el año 2018, periodo en el que la mortalidad alcanzó los 15 mil 574 decesos apuntan datos recabados por la Secretaría de Salud (STCONAPRA, 2019).

Existen diversos factores que pueden ocasionar este tipo de percances, como menciona la Comisión Nacional de Seguridad (2015), los siniestros viales son derivados en su mayoría por la persona a cargo del manejo del vehículo, alcanzando un porcentaje del 80% , 7% del total de accidentes han registrado derivarse por una falla en el vehículo, un 9% se le atribuye a factores externos como los ambientales y apenas un 4% del porcentaje total se debió a las imperfecciones de la carretera.

Con base en los datos capturados por el INEGI (2020) se puede apreciar en la Ilustración 10 que el automóvil es el tipo de vehículo que mayormente sufre accidentes viales, motivo por el cual el presente trabajo se enfocará en ellos.

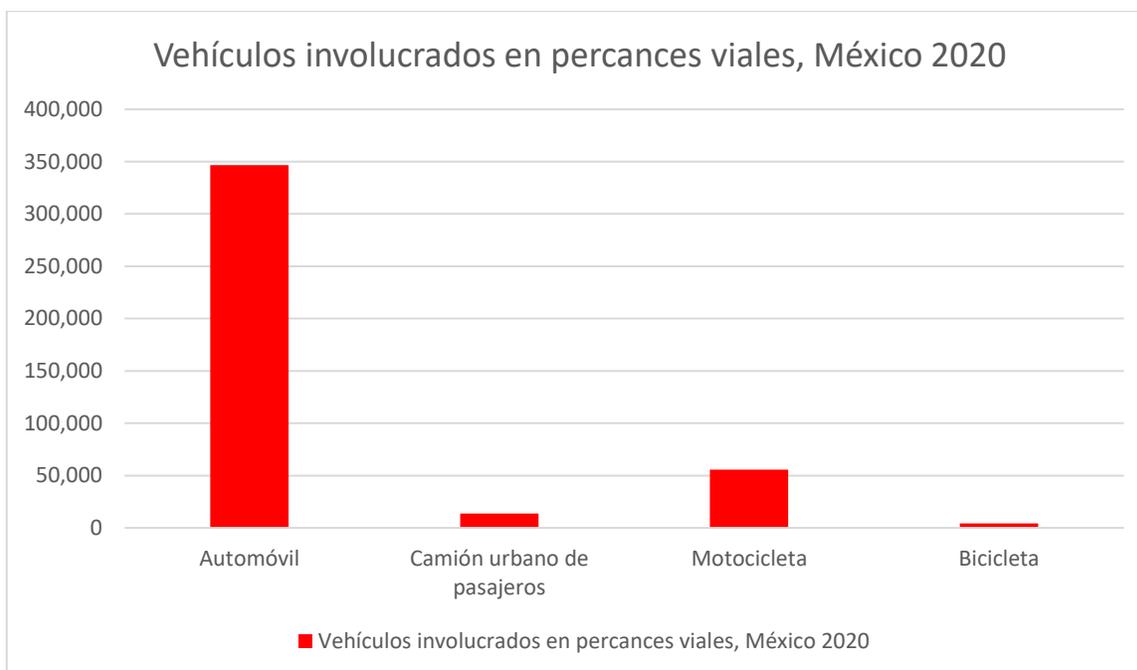


Ilustración 10 Accidentes de tránsito terrestre, vehículos involucrados obtenido de (INEGI 2020).

Como se apreció anteriormente, los accidentes de tránsito derivan de una inatención presentada por el conductor del automóvil, principalmente cuando se lleva a cabo una actividad distractora, las cuales son uno de los principales motivos que pueden desencadenar accidentes vehiculares en México. De acuerdo con la Fundación Carlos Slim (2016) los agentes distractores pueden presentarse de diferentes formas, a continuación se enlistan las cuatro clases en las que se agrupan:

- **Visuales:** corresponde al desvío de la atención del conductor hacia el camino para realizar alguna otra tarea como observar el celular o una pantalla de video integrada en el automóvil, cerrar los ojos por cansancio, admirar el paisaje, mirar una valla publicitaria o algún otro accidente ocurrido en el camino.

- Cognitivas: reflexionar sobre un tema de conversación, estar en estado de ánimo depresivo, el estrés causado por problemas externos del día a día, y el constante estado de alerta que genera la acción de conducir y estar atento ante cualquier peligro.
- Físicas: mientras se desarrolla alguna otra actividad que impida mantener las manos sosteniendo el volante del vehículo. Ejemplos de esto son: ir mandando mensajes de texto con un dispositivo móvil, manipular la radio del automóvil. El conductor también es propenso de perder la concentración del manejo mientras está ingiriendo bebidas o alimentos, enciende un cigarrillo, en algunos casos las mujeres deciden maquillarse mientras van conduciendo.
- Auditivas: el uso de los dispositivos móviles mientras se conduce ha ido en aumento, de igual forma el llevar música con volumen alto en la radio del automóvil representa un peligro que conlleva la disminución de la atención del conductor y su pérdida del tiempo de reacción ante cualquier imprevisto.

Como menciona Carro Pérez (2019), en la actualidad el comportamiento de los conductores de automóviles es un aspecto de vital importancia a nivel mundial para la prevención de la conducción distraída, lo cual puede derivar en accidentes, y es un tema poco estudiado en el contexto mexicano.

2.2 JUSTIFICACIÓN

2.2.1 Pertinencia

De acuerdo a Renault Eurofranceses de México (2021) se demuestra que la mayoría de accidentes vehiculares se pueden prevenir, ya que el 80% de ellos son originados por el conductor. La falta de pericia, el descuido provocado por alguna distracción y los efectos del alcohol y las drogas son los factores humanos más comunes que afectan a más del 50% de los conductores y que conducen a un accidente vial. Por lo tanto, es de vital importancia estudiar nuevas tecnologías capaces de analizar el estado de distracción para poder ayudar a mitigar este problema. Actualmente se han desarrollado sistemas que ya vienen integrados en vehículos de gama alta, enfocados en detectar algunos tipos de distracciones, sin embargo, estas propuestas son poco accesibles por su costo elevado, limitadas y se enfocan únicamente en detectar determinadas distracciones.

2.2.2 Relevancia

Con la presente investigación se busca implementar una red neuronal entrenada funcionando en un dispositivo móvil capaz de asistir a un conductor mediante la generación de alertas sonoras en la detección y alerta temprana de actividades distractoras, los cuales son de los motivos principales que pueden desencadenar accidentes de tráfico.

2.3 HIPÓTESIS

Es posible implementar un Sistema de Asistencia a la Conducción implementando redes neuronales convolucionales en un dispositivo móvil, capaz de detectar la conducción segura y nueve distracciones en conductores de automóviles: Enviando SMS con mano derecha, Llamada con mano derecha, Enviando SMS con mano izquierda, Llamada con mano izquierda, Manipulando la radio/GPS, Ingeriendo bebidas, Buscando algo en la parte de atrás, Maquillando/Arreglando el cabello, Hablando con pasajero.

2.4 OBJETIVOS

2.4.1 Objetivo general

Diseñar un Sistema de Asistencia a la Conducción para asistir a los conductores por medio de la detección de distracciones que sea capaz de generar alertas.

2.4.2 Objetivos específicos

- Definir el entorno del problema

- Analizar bases de datos existentes para realizar pruebas y estudiar su funcionamiento.
- Generar una base de datos propia capturando imágenes de conductores.
- Realizar el etiquetado de las imágenes.
- Realizar pruebas con diferentes modelos de redes y evaluaciones de su eficacia para reconocer distracciones.
- Implementar el sistema en un dispositivo móvil.

2.5 METAS

Para poder realizar los objetivos específicos se proponen las siguientes metas:

- Tener una clasificación de las distracciones usadas en bases de datos para comprender la identificación de distracciones usando imágenes.
- Identificar las distracciones que deban ser reconocidas por el sistema.
- Definir el lugar y posición del dispositivo móvil en el vehículo para la captura de la información.
- Definir el tipo de advertencia del sistema que se le dará al usuario para informarle cuando se detecte alguna distracción.
- Definir el tipo de red neuronal a utilizar en el sistema.
- Implementar la red neuronal con la base de datos modificada
- Desarrollar la interfaz del sistema en un dispositivo móvil.

2.6 LIMITACIONES DE LA TESIS

El tipo de dispositivo móvil tiene que ser compatible con el sistema operativo Android para celular. Otra de las limitaciones del proyecto serán las condiciones ambientales a las que se exponga la cámara del celular encargada de recibir la información del conductor. Estos cambios pueden llegar a afectar considerablemente la detección de distracciones, entre ellos se encuentran el exceso de luz provocado por el sol o por el alumbrado de otro vehículo y también la falta de luz cuando se conduzca de noche.

2.7 ORGANIZACIÓN DE LA TESIS

El presente trabajo de tesis está enfocado a la aplicación de la inteligencia artificial para la contribución a la solución de problemas sociales tales como la prevención de distractores que puedan ocasionar accidentes viales. Se propone un sistema de asistencia a la conducción entrenado con redes neuronales convolucionales adaptado a una aplicación para un dispositivo móvil, lo cual permitirá una fácil movilidad y utilización del sistema para los usuarios.

En el capítulo 3 se abarcan los fundamentos teóricos de la visión artificial, así como de sus limitaciones las cuales afectarán en la implementación y en el desempeño del sistema. Se presentan también conceptos referentes al aprendizaje profundo y a la inteligencia artificial, el cual es utilizado en este trabajo para el entrenamiento y clasificación dentro del sistema. El capítulo 4 engloba el procedimiento, la experimentación y las dificultades que se encontraron durante la búsqueda de un

modelo óptimo capaz de clasificar correctamente las distracciones deseadas. También se detalla el procedimiento necesario para poder adaptar y desplegar el modelo obtenido en un dispositivo móvil. El capítulo 5 abarca los resultados del sistema y a su vez son comparados con los antecedentes y finalmente en el capítulo 6 se encuentran las conclusiones obtenidas por el proyecto de investigación.

Capítulo III: Marco teórico

3.1 Visión artificial

La visión artificial se encuentra conformada por un conjunto de métodos encargados de hacer la captura del objeto mediante cámaras, llevar a cabo un preprocesamiento previo de las imágenes para su posterior análisis y mediante un proceso computacional poder realizar una interpretación numérica a partir de las imágenes tomadas del exterior.

Su objetivo es automatizar tareas reservadas hasta hace poco tiempo al ámbito humano en áreas como la seguridad, la industria, el comercio, la medicina, etc.

En la Ilustración 11 se puede observar un ejemplo de la captura de una imagen, en donde intervienen el objeto a analizar y procesar computacionalmente, el sistema encargado de digitalizar el objeto observado y es necesaria también la intervención de una fuente luminosa para que el sistema sea capaz de reconocer todas las características del objeto. De la misma manera en que nuestros ojos son susceptibles a la luz para poder distinguir correctamente lo que hay a nuestro alrededor, los sistemas de captura de imágenes también lo son, es por ello que el proceso se asemeja con el sentido de la vista en donde una vez obtenida la imagen pasa a ser procesada por el cerebro y por computadoras en el caso de la visión artificial (Azuela, 2013).

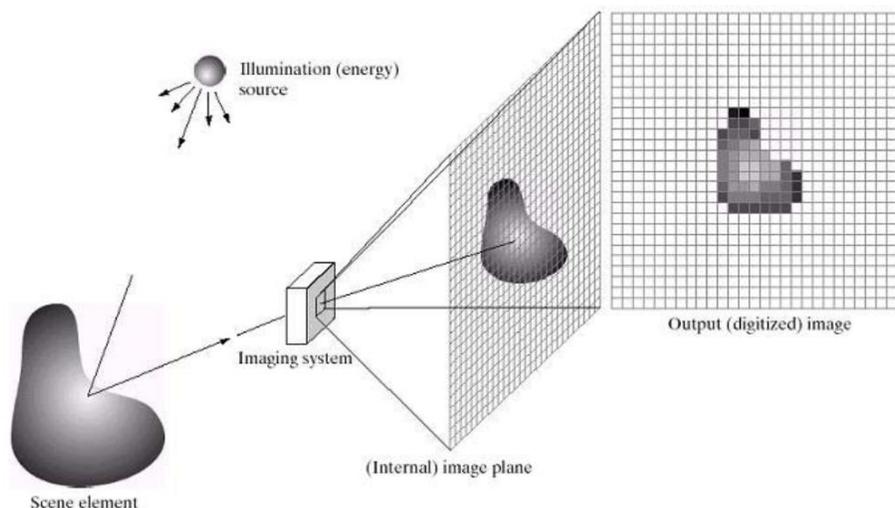


Ilustración 11 Ejemplo de la captura de una imagen y su procesamiento digital obtenido de Azuela, (2013).

Dentro del campo de la visión artificial nos podemos encontrar con dos vertientes, por un lado, se cuenta con los procesos tradicionales en los que es necesario aplicar laboriosos y costosos programas de extracción de características de las imágenes, procesos que actualmente son solventados por el uso del aprendizaje profundo y de las redes neuronales artificiales encargadas automáticamente de llevar a cabo la extracción de la información de las imágenes mediante la realización de entrenamientos. Estas arquitecturas de redes neuronales artificiales tienen la capacidad de clasificar imágenes en las clases para las que fueron entrenadas, en la ilustración 12 se puede apreciar un ejemplo de las capacidades con las que cuentan, entre ellas la clasificación y detección de objetos. Para la realización de estas tareas es necesario tener en primera instancia una base de datos con las imágenes y nombres (etiquetas) de lo que queremos detectar y clasificar, con estos datos, se

procede a realizar los entrenamientos de los modelos de aprendizaje para que el sistema sea capaz de realizar predicciones basadas en las características anteriormente aprendidas (Fabio & Manuela, 2012).

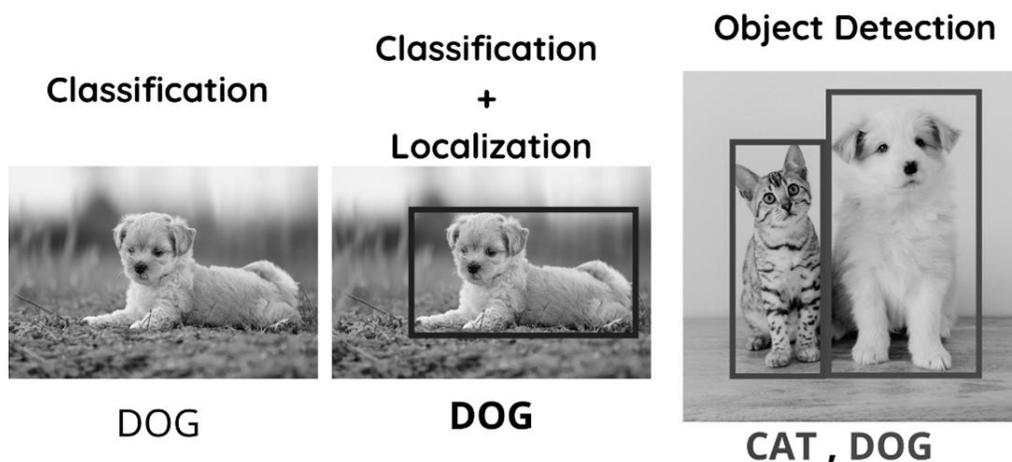


Ilustración 12 Ejemplo de clasificación de dos clases obtenido de (Fabio & Manuela, 2012).

3.1.1 Dificultades de la visión artificial

A pesar de los continuos avances que ha ido teniendo la visión artificial, se sigue enfrentando a uno de los mayores problemas para la clasificación de imágenes, la ambigüedad de las clases, es decir, cuando las imágenes de las clases a identificar son muy parecidas entre sí y el único rasgo con el que se diferencian suele ser muy difícil de detectar para las personas e incluso para los sistemas de visión, ejemplo de ello son las distracciones en conductores como se muestra en la Ilustración 13, ambas clases representan una distracción con el celular, parecidas y difícil de diferenciar (Goro & Ashish, 2007).



Ilustración 13 Clases parecidas entre sí

3.1.2 Variaciones en la iluminación

Las grandes variaciones en la luz pueden llegar a ocasionar pérdidas en la información de captura de las imágenes. En la Ilustración 14 se puede observar variaciones en la iluminación siendo la de en medio una iluminación uniforme recomendada.



Ilustración 14 Variaciones en la iluminación obtenido de (Goro & Ashish, 2007).

3.1.3 Movimiento

De igual forma las cámaras son susceptibles al movimiento que pueda presentarse tanto por el objeto analizado en cuestión, como por el que pueda sufrir la cámara que lo está enfocando. Esto puede representar errores en la captura de las imágenes o en la extracción de sus características. En la Ilustración 15 se puede notar en la imagen de la izquierda, una imagen capturada con un buen enfoque, mientras que en la imagen de la derecha se aprecia que la cámara sufrió un movimiento que le impidió capturar adecuadamente la información del exterior.

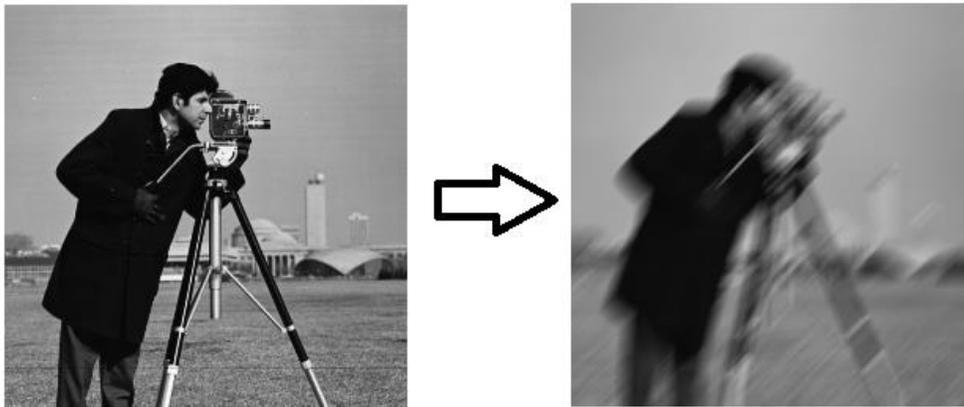


Ilustración 15 Desenfoco de la imagen por movimiento obtenido de (Goro & Ashish, 2007).

3.2 Imagen digitalizada

Las imágenes capturadas y digitalizadas son representadas mediante funciones de intensidad en 2D, representadas en el plano digital (Ilustración 16) mediante

coordenadas espaciales y su valor de intensidad en cada una de estas coordenadas (Murguía, 2007).

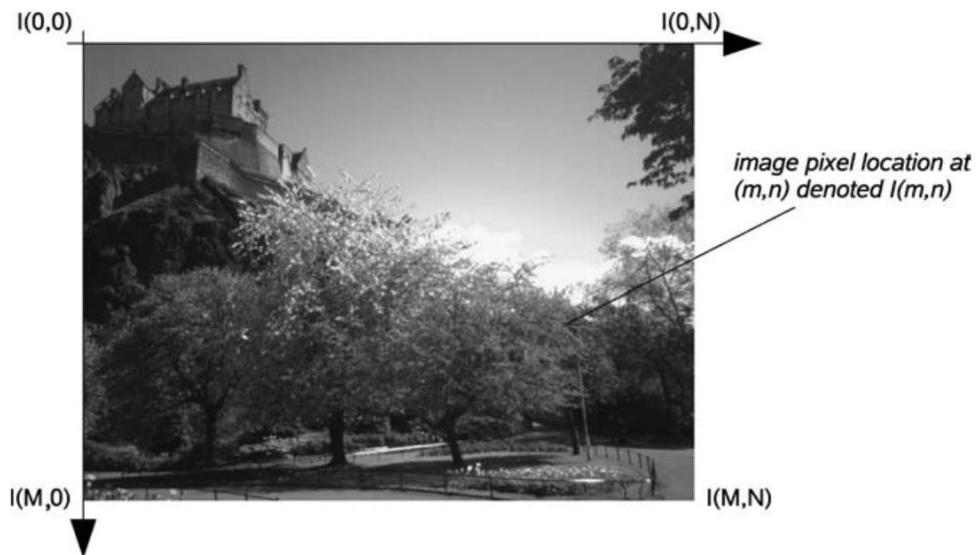


Ilustración 16 El espacio de coordenadas cartesianas 2-D de una imagen digital $M \times N$ obtenido de (Murguía, 2007).

3.3 Procesamiento digital de imágenes

Consiste en el tratamiento, el procesamiento y análisis de la información que se encuentra en una imagen digital. Como se muestra en la Ilustración 17, los objetivos y técnicas utilizadas en este proceso dependen en medida del tipo de aplicación y tiene como objetivo eliminar los efectos que puedan provocar una pérdida en la información

de la imagen. Mediante esta herramienta se pueden resaltar zonas específicas que se desean analizar (Chris & Toby, 2011).

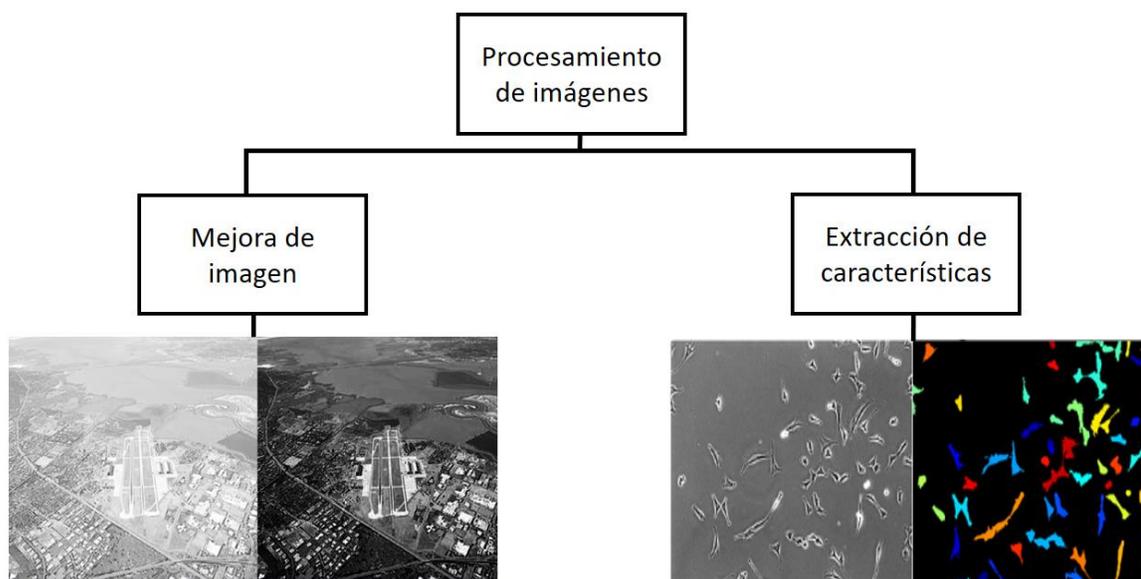


Ilustración 17 Mejora y extracción de características de la imagen obtenido de (Murguía, 2007).

3.3.1 Conversión de imágenes de 3 canales RGB a 2 canales escala de grises

Una imagen en escala de grises está conformada por un solo valor de intensidad, en cambio, una imagen a color cuenta con 3 intensidades pertenecientes a los canales rojo, verde y azul, que, al combinarlos, determinan el color final de cada pixel. De esta manera para poder calcularlos valores de la imagen en su escala de grises es necesario aplicar una ponderación a cada uno de los valores de los pixeles que componen los canales RGB de la imagen a calor (Ilustración 18). Lo comentado con anterioridad puede aplicarse por el método tradicional como se representa en la

ecuación de la Ilustración 18, o de igual forma, actualmente es posible utilizar librerías capaces de automatizar este proceso (Roberto & Juan, 2012).

$$G = 0.3 * (R) + 0.5 * (G) + 0.11 * (B)$$

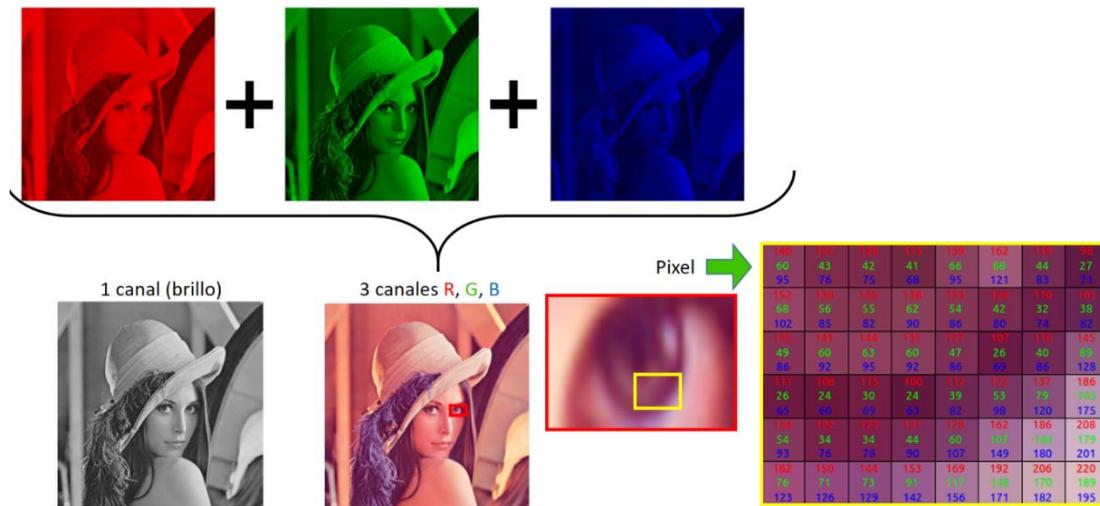


Ilustración 18 Conversión de imagen a color a escala de grises obtenido de (Roberto & Juan, 2012).

3.4 Inteligencia artificial

Es un tipo de tecnología aplicada a máquinas (Ilustración 19), para que éstas desarrollen tareas que implican capacidades similares a las que puede desarrollar el ser humano.

Pertenece a la rama de las ciencias computacionales, enfocadas en buscar la forma de emular el razonamiento humano para la rápida y óptima toma de decisiones,

este tipo de tecnología ha desarrollado varias vertientes como se puede observar en la Ilustración 19 (Max & Devedzic, 2004).

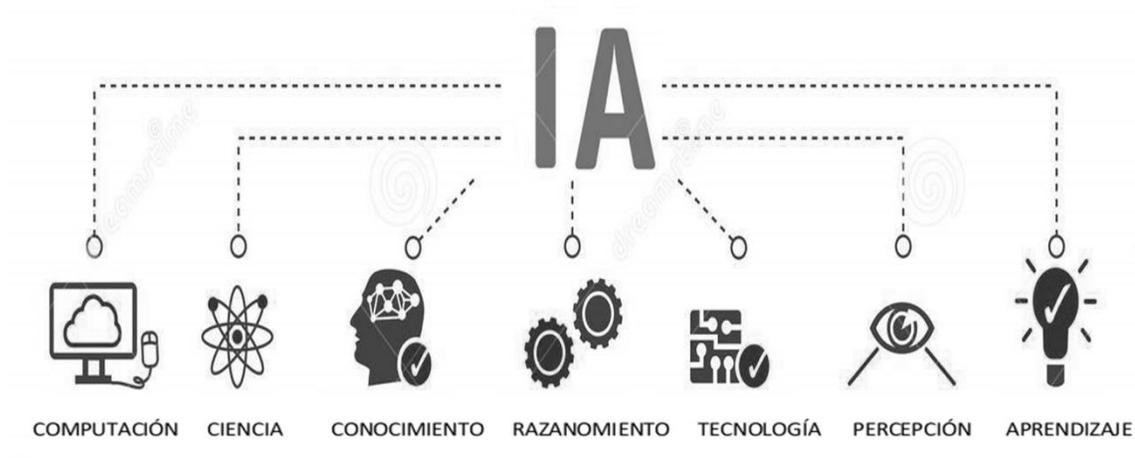


Ilustración 19 Características de la IA.

La inteligencia artificial trabaja mediante el uso de algoritmos, los cuales tienen la capacidad de, mediante un exhaustivo entrenamiento, poder obtener las características de los datos de entrada (imágenes, audio, video, bases de datos) y poder predecir un resultado para la entrada de nuevos datos que no estuvieron presentes en los entrenamientos. La gran ventaja que ha alcanzado la implementación de este tipo de tecnología ha sido que son capaces de manejar grandes cantidades de datos sin requerir de un descanso, y a diferencia de una persona común, el riesgo de cometer errores es significativamente pequeño (Konar, 2000).

El uso de esta tecnología es aplicable a casi todos los problemas que conlleve el análisis de datos y predicciones, a continuación, se enlistan aquellos campos que han tenido un crecimiento en la actualidad:

- El reconocimiento de objetos dentro de imágenes, la clasificación y predicción de clases de imágenes.
- El procesamiento de datos en empresas, hospitales o en gobierno, para un acceso a los datos más eficiente.
- En la industria automotriz ha tenido un gran impacto en el desarrollo de los vehículos autónomos y sus sistemas de navegación autónoma.

3.4.1 Detección de objetos en imágenes

La extracción de características propias de las imágenes es un método para la identificación de uno o múltiples objetos mediante la búsqueda de las características particulares de los objetos a detectar contenidos dentro de las imágenes, enfrentándose a problemas como la variabilidad del tamaño del objeto, los obstáculos (otros objetos) que dificulten la identificación de lo que se quiere distinguir, rotación y posición dentro de la misma imagen, por mencionar algunas.

La Ilustración 20 muestra el proceso para llevar a cabo la detección de objetos, en primer lugar es necesario contar con una cámara para poder capturar las imágenes del entorno en el que se encuentre el objeto, posteriormente se lleva a cabo un procesamiento para resaltar las características del objeto analizado y descartar toda aquella información que no sea necesaria, continuando con la extracción de características particulares que nos ayuden a desarrollar el sistema de reconocimiento (Yildirim, 2010).

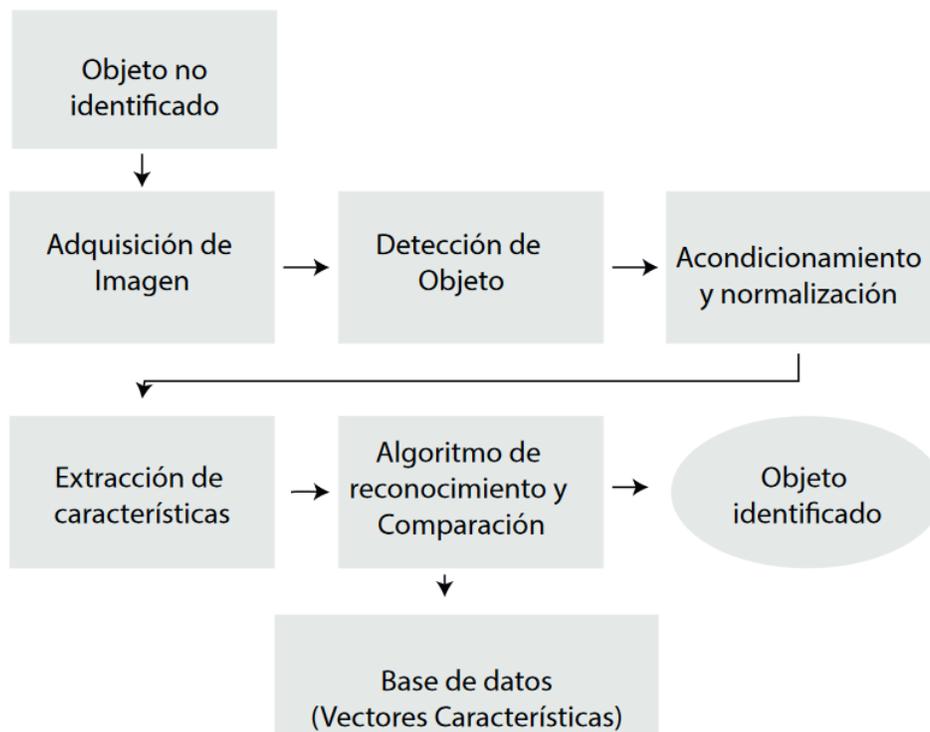


Ilustración 20 Modelo funcional del reconocimiento de imágenes obtenido de (Yildirim, 2010).

3.4.2 Computación afectiva

Uno de los grandes objetivos que persigue la Inteligencia Artificial es poder realizar actividades que sean lo más parecidas a las que realiza el ser humano, su capacidad de predecir resultados, diagnosticar fallas, distinguir y clasificar objetos, toma de decisiones confiables y rápidas y el análisis de emociones son algunas de ellas, si bien, las máquinas aún no han conseguido la capacidad que tiene el ser humano de poder manifestar sus emociones, han conseguido la capacidad de poder analizarlas y distinguirlas en imágenes y en texto.

El análisis de sentimientos por medio de imágenes tiende a reconocer y diferenciar las emociones por medio de la extracción de las características que presentan las variaciones del rostro de las personas, por el contrario, el análisis por medio de texto distingue las palabras utilizadas para una buena crítica o una negativa, logrando distinguir los gustos de las personas. El análisis de sentimientos ha cobrado tal importancia en la actualidad para poder predecir y sugerir recomendaciones de productos a las personas, analizar la respuesta del consumidor ante la salida de algún nuevo producto o película e incluso para la preservación de la vida, mediante sistemas encargados de monitorear la salud los pacientes o sistemas de seguridad que alerten algún estado inconveniente o pueda llegar a perjudicar a la persona.

Esta rama de la tecnología involucra varios campos interdisciplinarios, entre ellos, la programación, el análisis psicológico de los sentimientos y sus repercusiones en las personas, la detección de objetos y el aprendizaje automático.

Desde sus orígenes en 1995 hasta la actualidad ha ido incorporando mejoras, tal es el caso de la integración de los modelos de aprendizaje profundo (Or, 2011).

3.5 Aprendizaje automático

El aprendizaje automático es un subconjunto de la inteligencia artificial, enfocada en asimilar la capacidad que tiene el ser humano de aprender, mediante algoritmos de entrenamientos en el caso de los modelos de aprendizaje, la capacidad de detectar objetos, distinguir y diferenciar entre clases. Estos modelos requieren de

grandes conjuntos de datos (bases de datos) de imágenes, audio, video o texto para poder llevar a cabo un aprendizaje de que es lo que se está queriendo analizar, extraer sus características y poder ejecutar una predicción en la salida (Alpaydin, 2020).

La ventaja que presentan estos algoritmos es que no requieren tanto la intervención humana para adquirir conocimiento, basta con ajustar una serie de hiperparámetros necesarios para los modelos, diseñar el modelo de la red, y de esta manera el sistema se encargará de aprender por su cuenta (Burch, 2001).

El aprendizaje automático se puede categorizar en tres subconjuntos como se puede apreciar en la Ilustración 21.

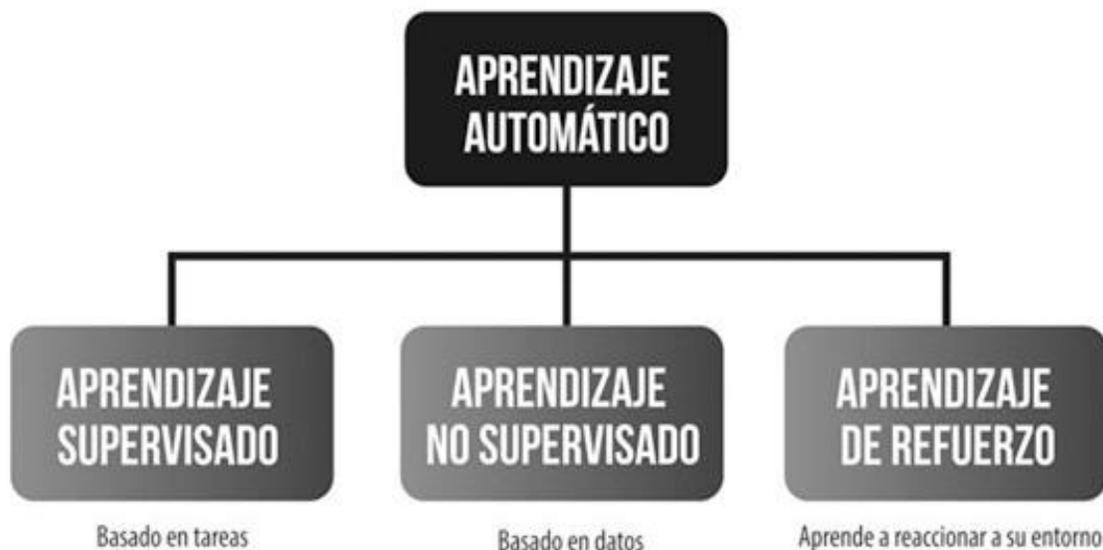


Ilustración 21 Subconjuntos del aprendizaje automático.

3.5.1 Aprendizaje supervisado

La vertiente del aprendizaje supervisado (Ilustración 22) requiere como punto de partida un conjunto de datos etiquetados, es decir, proveer de ejemplos con la respuesta que esperamos obtener a la salida de nuestro modelo para que este mismo logre extraer mediante el entrenamiento de algoritmos, las características únicas con las que se pueden diferenciar cada una de las clases existentes en el problema. Existen dos subconjuntos dentro del aprendizaje supervisado, si lo que se requiere obtener es una predicción que nos dé a la salida un número entero para determinar a qué clase pertenece el dato de entrada a la red, se estará tratando un problema de clasificación, por el contrario, si la respuesta a predecir es un número real se estará tratando un problema de regresión (Harrington, 2012).

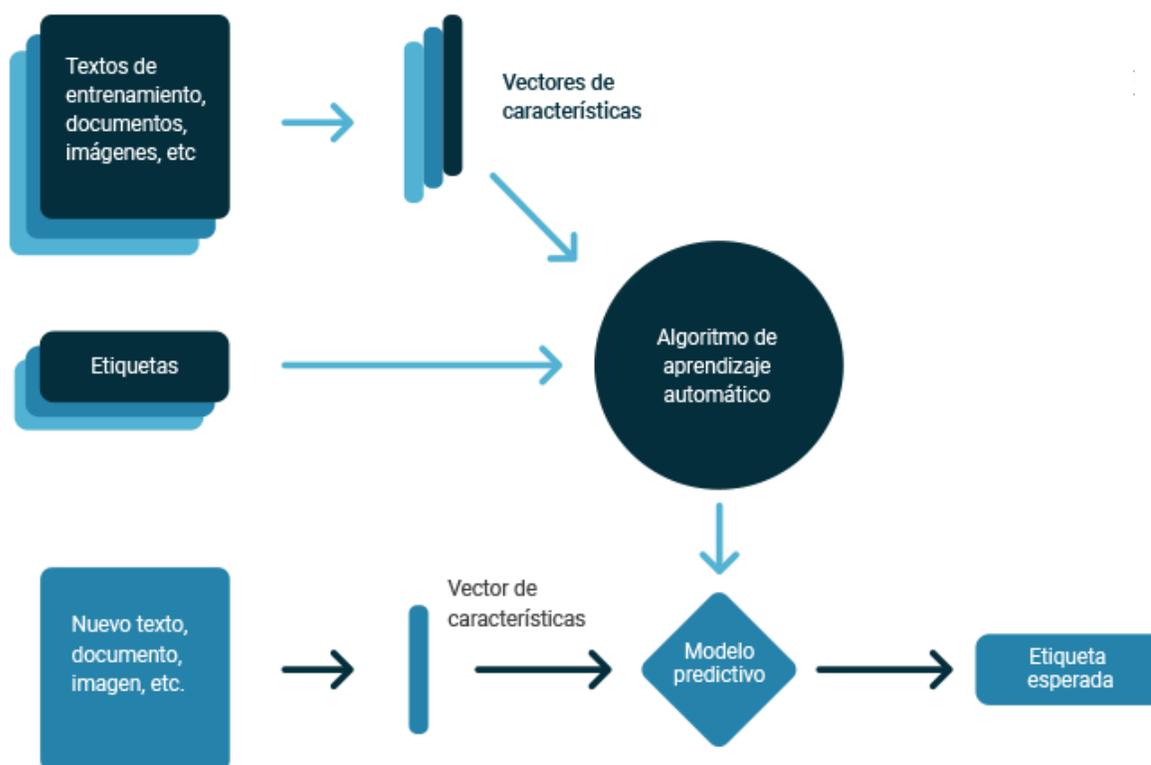


Ilustración 22 Modelo de aprendizaje supervisado obtenido de (González, 2018).

3.5.2 Aprendizaje no supervisado

En el aprendizaje no supervisado (Ilustración 23) los algoritmos no usan ningún dato etiquetado u organizado previamente para indicar cómo tendría que ser categorizada la nueva información, es decir, no se tiene con antelación la respuesta que queremos obtener del sistema, sino que este mismo tendrá la tarea de encontrar las relaciones existentes entre todo el conjunto de datos disponible y poder agrupar los datos en agrupaciones similares, que compartan las características anteriormente encontradas por el sistema en cada uno de los grupos (González, 2015).

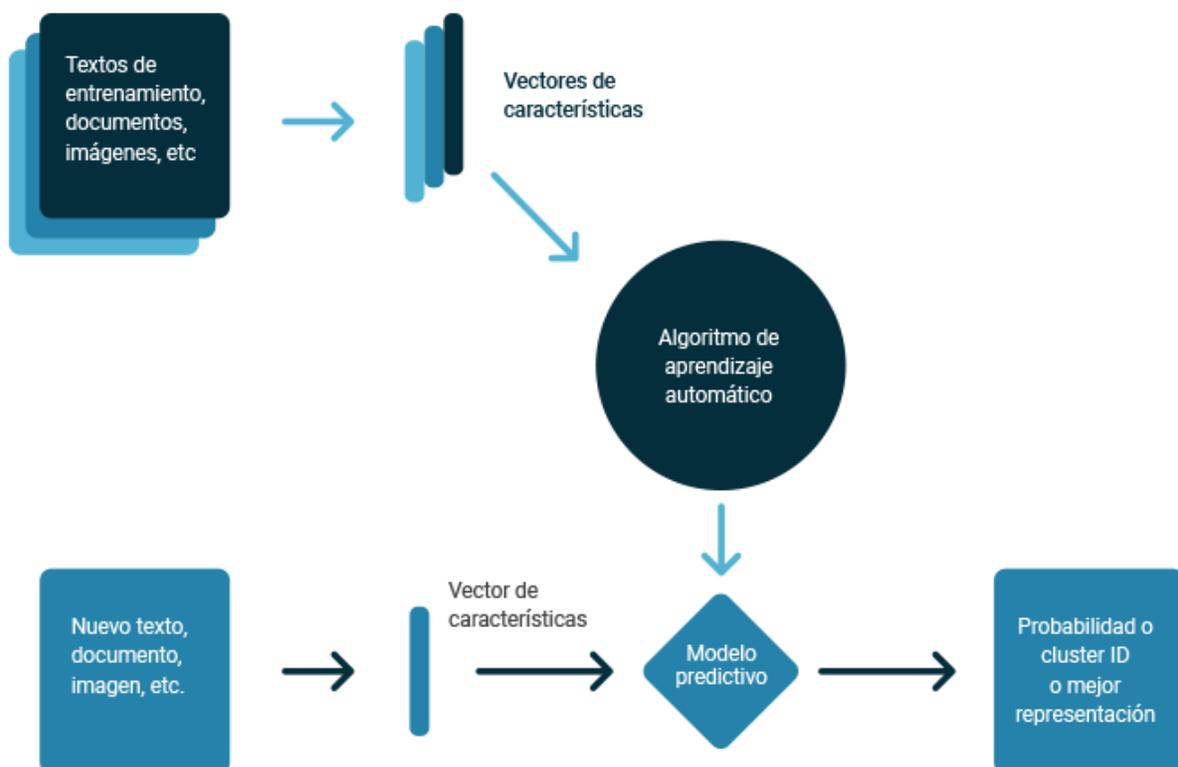


Ilustración 23 Aprendizaje no supervisado obtenido de (González, 2018).

3.5.3 Aprendizaje por refuerzo

En los modelos de aprendizaje por refuerzo (Ilustración 24), los algoritmos aprenden de la experiencia. En otras palabras, tenemos que darles “un refuerzo positivo” cada vez que aciertan, y negativos cuando se equivocan, normalmente son utilizados cuando se cuenta con una base de datos robusta (gran número de datos) y muy pocos de ellos se encuentran etiquetados, la forma en la que estos algoritmos aprenden se puede comparar con la de los perros cuando les damos “recompensas” al aprender a sentarse.



Ilustración 24 Modelo de aprendizaje por refuerzo obtenido de (González, 2018).

En el desarrollo de este trabajo se hará uso de la técnica de aprendizaje supervisado el cual es el más utilizado en la solución de trabajos de clasificación de

imágenes debido a que este enfoque utiliza un conjunto de datos etiquetados utilizados en algoritmos para clasificar datos y predecir salidas con precisión.

3.6 Redes neuronales artificiales

Las redes neuronales artificiales han sido la manera que ha obtenido la inteligencia artificial y el aprendizaje automático de tratar de emular y representa el funcionamiento que tiene el cerebro humano para poder aprender con la información que recibimos y poder tomar decisiones basadas en dicha información analizada (Arbib, 2019).

La estructura de las redes neuronales es simple, se agrupan por capas interconectadas entre sí para poder ir transmitiendo la información entre cada una de ellas, tal y como procesan la información del mundo real las neuronas contenidas en el cerebro humano. Para su análisis las capas suelen categorizarse en tres grupos, las capas de entrada son aquellas encargadas de recibir los datos de entrada a la red, la de en medio son comúnmente denominadas capas ocultas, son las encargadas de llevar a cabo los procesos de obtención de características de los datos por medio de filtros convolucionales y finalmente se encuentra la capa de salida, que es una combinación de neuronas totalmente conectadas entre sí, encargadas de realizar la tarea de predicción del dato de entrada en base a las características que se obtuvieron previamente en el entrenamiento (Graupe, 2019).

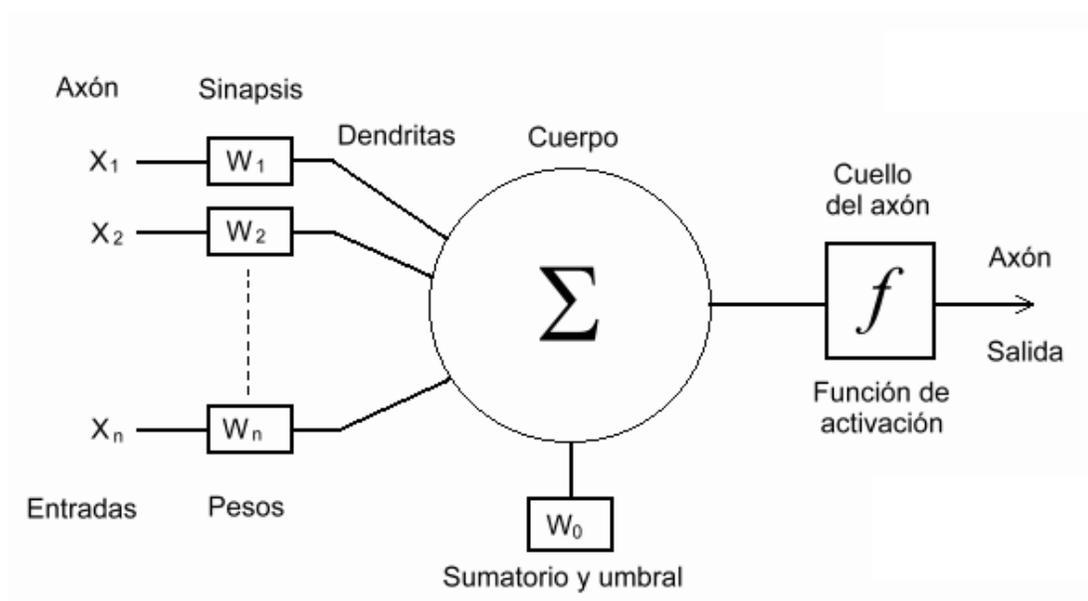


Ilustración 25 Modelo estándar de una neurona artificial obtenido de (Berzal, 2019).

En la Ilustración 25 se muestra una representación de los componentes de una neurona artificial, compuesta por las entradas cuya función es recibir los datos de entrada o de las capas anteriores, pesos que serán ponderados junto con los valores de entrada para la obtención de las características y una función de activación que nos permitirá aprender patrones a través de los datos y delimitar la respuesta que queremos obtener en la salida que la red neuronal (Berzal, 2019).

3.6.1 Redes artificiales monocapa

Una red monocapa (Ilustración 26) es la forma más simple de un modelo de aprendizaje, la capa realiza propagaciones hacia adelante para la obtención de

características. Estas redes se utilizan para el tratamiento de problemas que no sean complejos debido a la única capa disponible (Michie, Spiegelhalter, & Taylor, 1994).

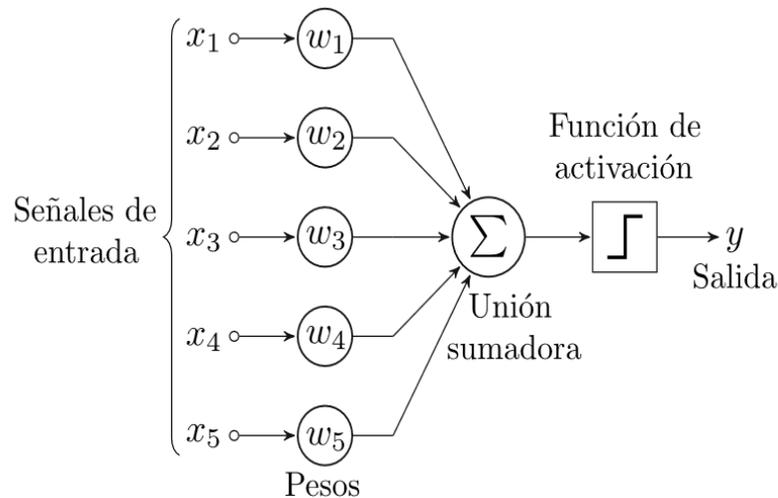


Ilustración 26 Modelo de red neuronal monocapa obtenido de (Michie, Spiegelhalter, & Taylor, 1994).

3.6.2 Redes neuronales artificiales multicapa

A diferencia de las redes neuronales monocapa, las redes multicapa cuentan con unas capas ocultas en su modelo encargadas de aplicar filtros convolucionales a los datos para la obtención de las características y posterior predicción en la capa de salida. Una red se considera multicapa cuando esta cuenta con más de una capa, tal y como se muestra en la Ilustración 27 (Haykin y Cols., 2009).

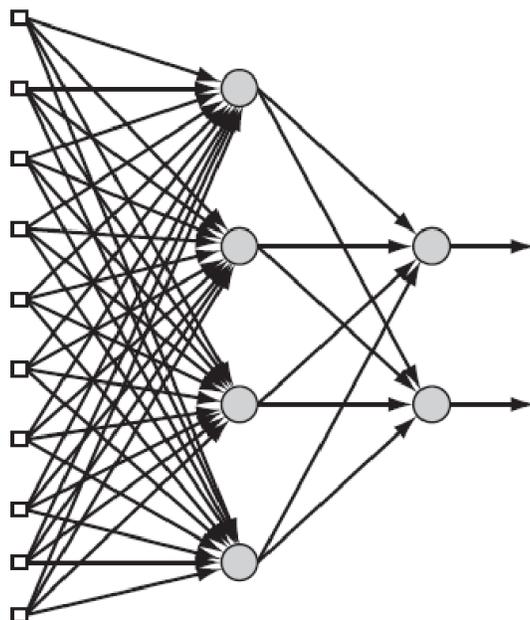


Ilustración 27 Modelo de una red neuronal multicapa obtenido de (Simon, 2009).

Dentro de los problemas a los que se enfrentan las redes, se encuentran los linealmente separables, aquellos problemas ideales en los que existe mucha diferencia entre las clases y es más fácil encontrar una solución que se adapte al problema, por otro lado, se encuentran los que no son linealmente separables, problemas más complejos que requieren una mayor cantidad de entrenamientos puesto que las características de las clases son más parecidas entre sí y es más fácil tener errores en la salida del clasificador, este tipo de configuración de red neuronal puede ser utilizada para resolver problemas que no son linealmente separables. En la Ilustración 28 se observa un ejemplo de un clasificador lineal y uno no lineal, como se explicó con

anterioridad, los problemas no lineales tienden a ser más complejos en buscar su solución puesto que las clases son muy similares entre sí (Rabunal y Dorrado, 2005).

De esta manera, las redes neuronales multicapa son capaces de resolver tanto los problemas lineales como los no lineales al contar con más capas que las redes monocapa.

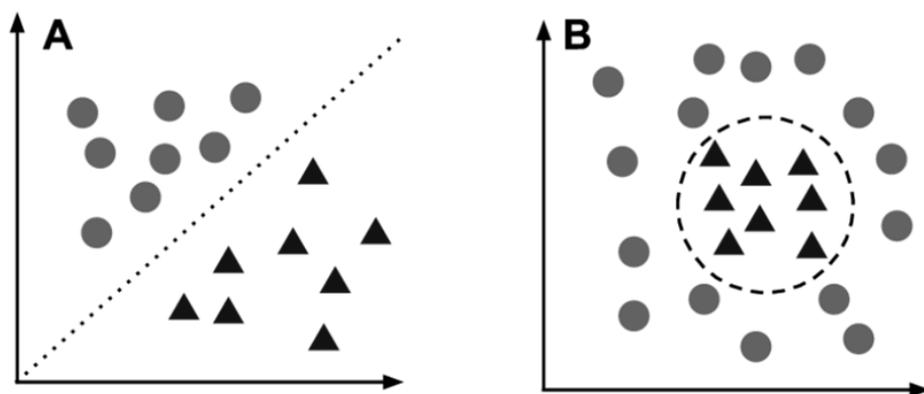


Ilustración 28 Clasificador lineal y no lineal.

3.6.3 Redes neuronales artificiales profundas

Las redes neuronales profundas (Ilustración 29) son un tipo de red que involucra un mayor número de capas ocultas que las redes multicapa, teniendo la capacidad de poder resolver problemas mucho más complejos. El propósito principal de una red de este tipo es recibir un conjunto de entradas, realizar cálculos progresivamente complejos en ellas y dar salidas para dar predicciones más acertadas a los problemas a los que se enfrenta (López y Fernández, 2008).

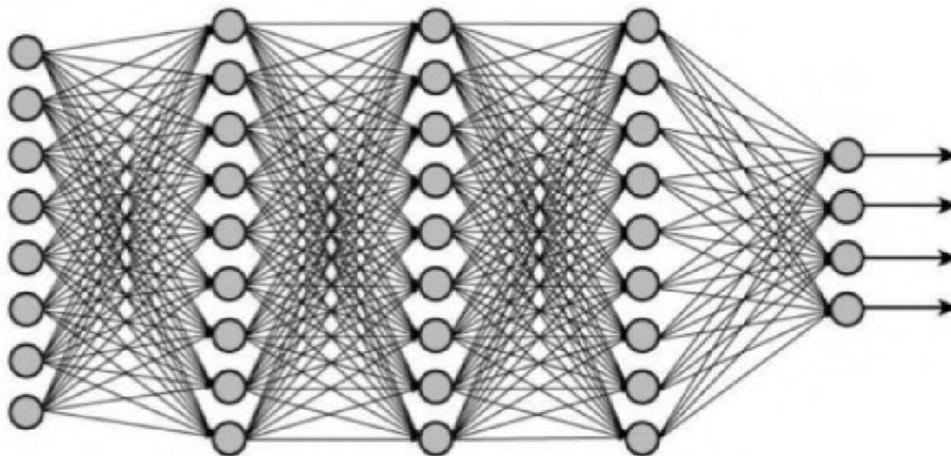


Ilustración 29 Modelo de una red neuronal artificial profunda obtenido de (López & Fernández, 2008).

3.7 Redes neuronales convolucionales

Las redes neuronales convolucionales son un subconjunto de redes neuronales profundas modeladas para operar con imágenes para la solución de tareas como el reconocimiento e identificación de objetos o la clasificación multiclase. Una etapa inicial convolucional y una etapa posterior de clasificación conforman su arquitectura.

La principal tarea de la etapa convolucional es la aplicación de filtros convolucionales a las imágenes de la base de datos en las capas ocultas de la red, poder encontrar las principales características que diferencian a los datos de cada una de las clases y de esta manera hacer predicciones para asignarle una clase a los nuevos datos de entrada (Behnke, 2003).

Durante la última etapa, la de clasificación, se realiza un aplanado de los datos para posteriormente introducirlos en capas totalmente conectadas en las que el número de neuronas de la última capa será equivalente al total de clases a predecir por el modelo.

3.7.1 Kernel

En las redes convolucionales, el kernel se define como un filtro que es utilizado para encontrar relaciones particulares entre el conjunto de datos y es aplicado en cada una de las capas de la red mediante convoluciones entre el kernel y las imágenes, variando en dimensiones tras cada capa. Este filtro es normalmente utilizado para la detección de texturas, bordes, entre otros.

3.7.2 Etapa convolucional

En matemáticas la convolución es una operación efectuada entre dos funciones cuyo resultado será una tercera función, interpretada como una versión filtrada de una de las dos funciones convolucionadas, en las redes, los filtros convolucionales son operaciones escalares aplicadas entre un kernel y los píxeles de las imágenes del conjunto de datos (Berzal, 2019).

En la Ilustración 30 se muestra el ejemplo del mapa de características de una imagen “I” a la cual se le aplicará un filtro convolucional “k” de tamaño 3×3 y al final se cuenta con el nuevo mapa de características “S” resultado de la convolución. La aplicación de los kernels convolucionales en imágenes sirve para resaltar o descartar propiedades importantes en el conjunto de datos, existen diferentes tipos de kernels, de diferentes dimensiones, especializados en adquirir características particulares (Lindblad & Kinser, 2005).

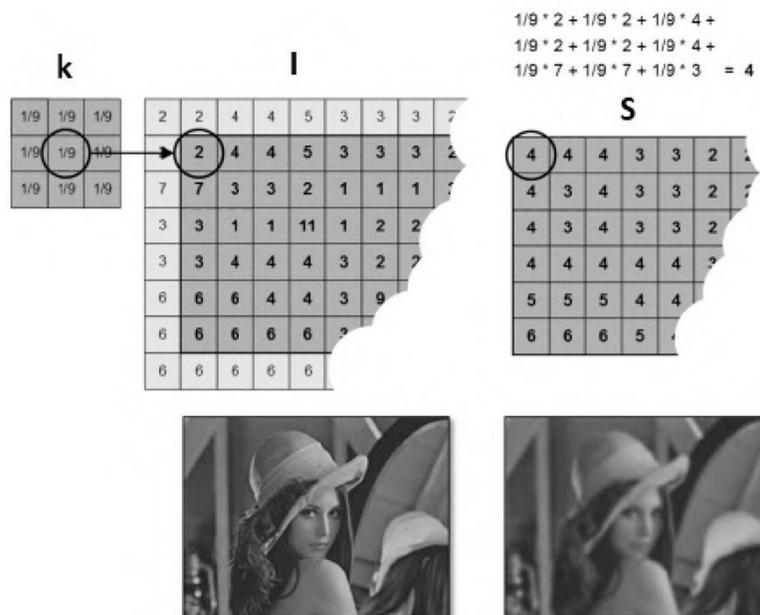


Ilustración 30 Ejemplo de la convolución aplicada a una imagen obtenido de (Berzal, 2019).

3.7.3 Capa de pooling

Los filtros *pooling* tienen como objetivo reducir la dimensionalidad de los mapas de características obtenidos después de realizar las operaciones convolucionales entre dos capas de la red en las imágenes de la base de datos, lo que se busca es poder reducir el número de total de parámetros entrenables para la red y poder quedarse únicamente con los parámetros más sobresalientes entre cada capa en la que es utilizado este filtro. La forma más común es “*Max-Pooling*”, el cual obtiene el pixel con el máximo valor de cada región.

3.7.4 Aplanado de los datos

El aplanado de los datos es un paso necesario que se da antes de pasar los mapas de características a través de las capas totalmente conectadas de la red, mismas que se encargan de otorgar los resultados de salida (predicciones de la red), estos mapas son acomodados de múltiples mapas matriciales a solo un vector unidimensional (Ilustración 31).

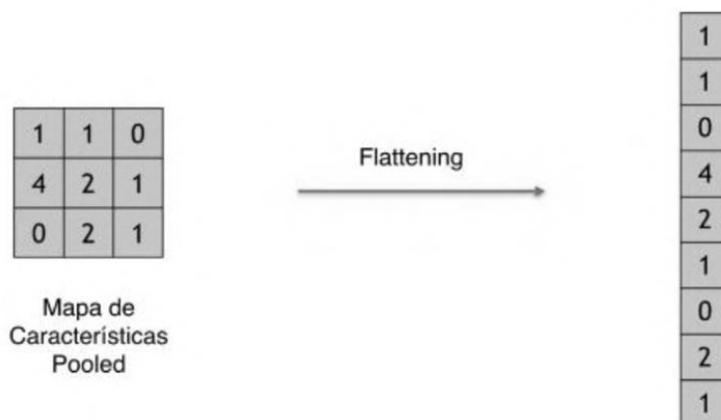


Ilustración 31 Aplanado de los datos.

3.7.5 Etapa de clasificación

Una vez teniendo el mapa de características convertido en un vector unidimensional, es introducido a la última etapa, la de clasificación, en las cuales están involucradas capas de redes neuronales totalmente conectadas (Ilustración 32), aquí se llevan a cabo las predicciones de los datos hacia sus clases correspondientes (Gomilla, 2019).

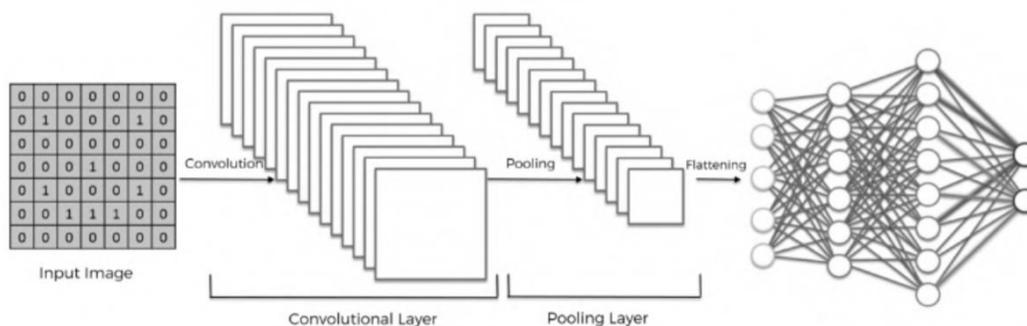


Ilustración 32 Ejemplo de las etapas de una red neuronal convolucional obtenido de (Gomilla, 2019).

3.7.6 Tasa de aprendizaje

La tasa de aprendizaje o learning rate en inglés, es lo que indica la importancia que le damos al error al actualizar cada peso, es decir, la rapidez o cómo de abruptos son los cambios en los pesos de las neuronas. Así una tasa de aprendizaje muy alta, hará que los cambios en los pesos sean muy grandes de una iteración a otra, lo que

tiene el problema de que se pueda llegar a pasar la respuesta ideal, la desventaja de una tasa de aprendizaje baja es que se eleva el costo computacional y se corre el riesgo de no obtener un buen modelo, este hiperparámetro es ajustable a criterio del programador (DeepChecks, 2022).

3.7.7 Tamaño del lote

El tamaño del lote o batch size en inglés se utiliza cuando el conjunto de datos es muy grande y este se suele agrupar por lotes más chicos para su entrenamiento, el cual se refiere al número de muestras con las que se va a trabajar antes de actualizar los parámetros internos del modelo y mejorarlo, este hiperparámetro es ajustable a criterio del programador (HUAWEI, 2023).

3.7.8 Número de épocas

Una época o epoch en inglés, es cuando todos los datos de entrenamientos se usan a la vez y se define como el número total de iteraciones de todos los datos de entrenamiento en un ciclo para entrenar el modelo de aprendizaje profundo. El número de épocas se considera un hiperparámetro, porque determina el número de veces que el algoritmo de entrenamiento procesará todo el conjunto de datos, este hiperparámetro es ajustable a criterio del programador (HUAWEI, 2023).

3.8 Transferencia de aprendizaje

La transferencia de aprendizaje o transfer learning en inglés, es un método del aprendizaje profundo en el que un modelo desarrollado para una tarea en concreto se puede utilizar como punto de partida para el desarrollo de otro modelo con una tarea similar a la cual fue entrenada (MATLAB, 2023). Este tipo de modelos requiere de una gran cantidad de datos para ser entrenada en la clasificación de miles de clases y de la misma forma requiere grandes recursos computacionales y tiempo para su correcto entrenamiento.

Las ventajas que tienen los modelos preentrenados frente a los modelos desarrollados desde cero es que pueden adaptarse a una base de datos con un problema similar y además:

- Tiene la factibilidad de que la base de datos utilizada puede contener una reducida cantidad de datos.
- Al utilizar la red preentrenada con los pesos aprendidos durante su entrenamiento, es posible reducir el tiempo de entrenamiento debido a que se cuenta con una menor cantidad de parámetros entrenables y la red solo busca “adaptarse” a los datos en su última etapa de clasificación.
- Los modelos disponibles son entrenados por empresas o grupos de investigación con un alto poder de cómputo.

Los modelos preentrenados utilizados en la transferencia de aprendizaje son desarrollados para tareas específicas en la clasificación de imágenes, derivado de sus

extensas capas profundas en donde extraen características principales en las imágenes es que tienen la ventaja de poder adaptarse a tareas similares para las que fueron entrenadas primordialmente, sin embargo, es posible “descongelar” capas ocultas de estos modelos para poder experimentar con ellas, con la desventaja de entre más capas se “descongelen” mayor serán los parámetros a analizar en los entrenamientos y mayor será el tiempo requerido para ello, incrementando el costo computacional.

3.9 Sistema ADAS

Los Sistemas Avanzados de Asistencia al Conductor, ADAS por sus siglas en inglés (*Advanced Driver Assistance Systems*) son sistemas en donde el vehículo es totalmente manipulado por un conductor, sin embargo, cuentan con un sistema de asistencia encargado de monitorear el entorno (algunos centrados en monitorear al conductor y otros más complejos encargados de monitorear las líneas delimitadoras del carril de la carretera o la dirección del volante) para prevenir conductas peligrosas que pueden poner en peligro la integridad física de los pasajeros.

Capítulo IV: Metodología y desarrollo

Para el desarrollo del proyecto se propone la metodología de investigación tipo cascada, la cual está basada en dividir el proyecto en distintas fases secuenciales y en resolver cada una de las etapas de forma consecutiva (Álvarez, 2015). Este tipo de metodología es indispensable para crear o actualizar software de calidad que cumpla con los requisitos del usuario, sus principales ventajas y desventajas se pueden apreciar en la Tabla 1:

Tabla 1 Metodología en cascada ventajas y desventajas

Ventajas	Desventajas
El modelo es simple y fácil de usar	Retrasa las pruebas para la última mitad
El proceso es bastante predecible	Se dificulta ir atrás en el proceso
Usa una estructura clara	Carece de flexibilidad ante eventos inesperados
Determina el objetivo final rápidamente	
Transmite bien la información	

Esta metodología es muy utilizada en el desarrollo de trabajos de software ya que contempla un orden riguroso en el desarrollo de sus etapas, en otras palabras, cada una de las ellas debe esperar la terminación de su etapa anterior. Debido a la rigurosidad de la metodología se realizó una modificación a la misma, con la finalidad de obtener un óptimo acoplamiento, lo cual es necesario debido al constante

entrenamiento que requieren las redes neuronales en la búsqueda de los hiperparámetros necesarios para obtener un modelo óptimo.

En la Ilustración 33 se detallan las fases a seguir para la realización del presente proyecto de investigación y se muestra en la etapa 5 la modificación realizada a la metodología por medio de un proceso iterable, el cual está encargado de realizar el continuo entrenamiento de la red neuronal.

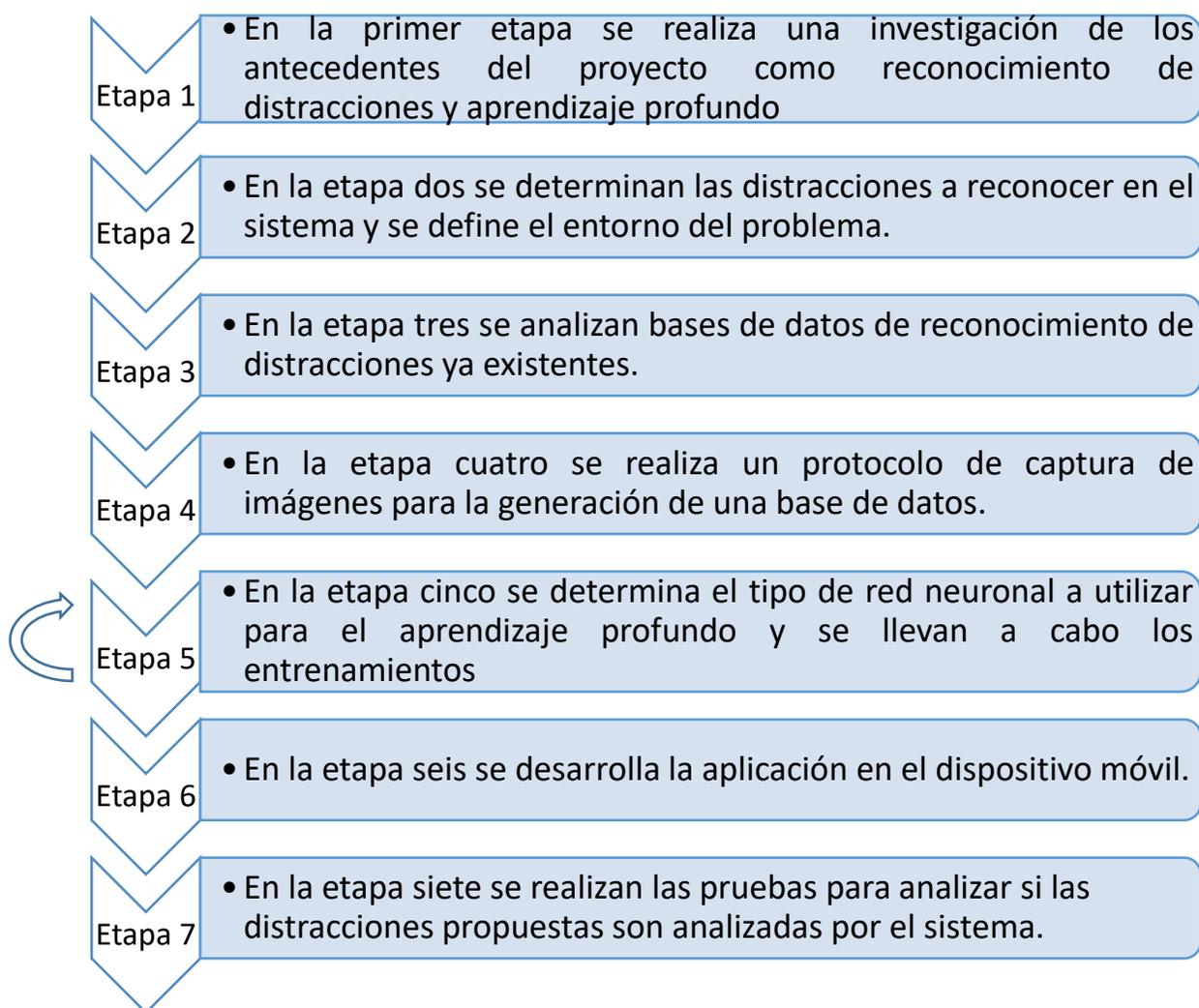


Ilustración 33 Metodología en cascada.

4.1 Etapa 1: Detección de distracciones en los conductores

El objetivo general del proyecto es el desarrollo de un sistema de visión artificial para asistir a conductores de automóviles por medio de la detección de distracciones mediante la evaluación y adaptación de bases de datos enfocadas a la detección de acciones en los conductores y el uso de técnicas de aprendizaje profundo enfocándose en las características fisiológicas de los conductores del sur de nuestro país para la generación de la base de datos.

4.2 Etapa 2: Definición de distracciones para el proyecto

La distracción durante la conducción se presenta ante sucesos que pueden ocurrir dentro de los pensamientos del conductor (distracciones cognitivas) o también cuando este mismo ejecuta acciones ajenas a la conducción lo que provoca el desvío de su atención. El atender una o varias acciones extras aparte de la conducción conlleva tarde o temprano al deterioro de las actividades realizadas, entre ellas la conducción.

El uso de los dispositivos móviles durante la conducción es uno de los tipos de distracciones más estudiadas, pues se ha demostrado que el uso del celular es capaz de absorber gran parte de nuestra capacidad de atención. Son muchas las personas que deciden leer o responder un mensaje de texto recién llegado a su celular mientras conducen, otras manipulan el celular para llamar y recibir llamadas haciendo que se desvíe la atención de la carretera (Protección Civil, 2021). De acuerdo con estudios de

la Dirección General de Tráfico (2020) indican que al utilizar el dispositivo móvil durante la conducción se genera un deterioro en la capacidad de atención, mismo que provoca la pérdida de la apreciación de los señalamientos de tránsito de hasta un 40% y el tiempo que se requiere para reaccionar ante algún imprevisto aumenta. El atender una llamada telefónica, escribir mensajes de texto o manipular la radio o el GPS, incrementa el riesgo de sufrir un accidente entre 5 y 10 veces más de lo normal, el tiempo promedio en que la vista se desvía para realizar estas acciones es de aproximado de 5 segundos y yendo a una velocidad urbana de 50 km/h equivaldría a recorrer más de 50 metros (alrededor de media cuadra) a ciegas. La Ilustración 34 muestra el tiempo aproximado que le toma a un conductor realizar alguna acción distractora y la distancia que recorre el vehículo sin que el conductor preste la atención necesaria a la carretera yendo a una velocidad de 50 km/h.

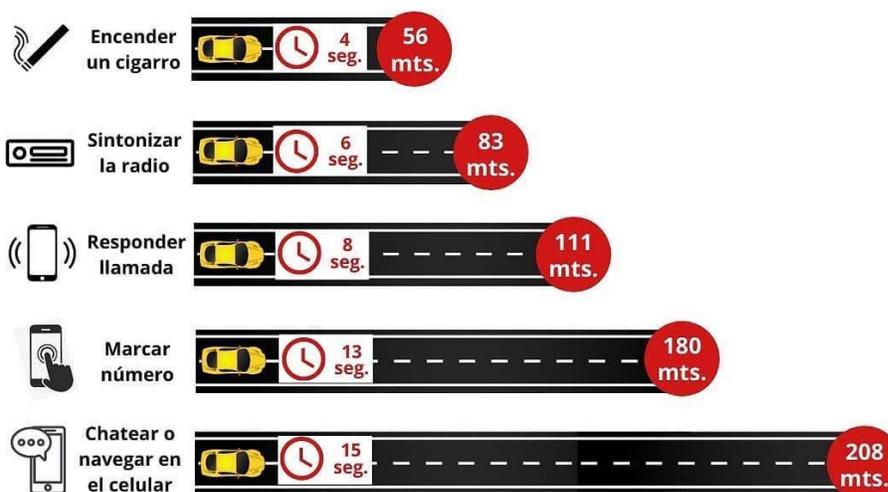


Ilustración 34 Tiempo y distancias recorridas por distracciones obtenido de (Dirección General de Tráfico de España, 2023).

Es preciso señalar que otro de los inconvenientes que involucran las distracciones es la disminución del tiempo de reacción ante algún imprevisto ya que luego de que el conductor se percata de algún peligro y procede a intentar frenar el automóvil lo más rápido posible, éste todavía tiende a recorrer una cierta distancia requerida para poder reducir su velocidad y parar el vehículo por completo y de una forma segura, dado que a mayor velocidad mayor será la distancia de frenado requerida, eso sin contar las condiciones de un día lluvioso en donde es más difícil llevar a cabo un frenado rápido. La Ilustración 35 es una representación a diferentes velocidades de la distancia que el automóvil requiere recorrer para realizar un frenado de emergencia.



Ilustración 35 Distancia de frenado ante imprevistos obtenido de (Instituto Mexicano del Transporte, 2022).

Aparte de desviar la atención al atender otras actividades, se está reduciendo aún más la capacidad de respuesta ante algún incidente al no contar con ambas manos al volante, resaltando que los conductores que están apartando una o ambas manos

del volante para realizar alguna acción fuera de la conducción están incumpliendo con el artículo 85 del reglamento de tránsito nacional, el cual se cita a continuación.

“Artículo 85: Todos los conductores de vehículos de motor deberán llevar asido firmemente con ambas manos, el control de dirección del vehículo” (SEMOVI, 2022).

Es de gran riesgo el buscar objetos que se encuentran en los asientos traseros del vehículo, al voltear tan solo 3 segundos para buscar algo puede provocar la invasión del carril contrario (Gobierno del estado de Jalisco, 2022).

Conducir rápido cuando se tiene la urgencia de llegar a un lugar no es recomendable ya que puede provocar la disminución de reaccionar ante algún peligro derivado de la urgencia de llegar al destino. Actualmente es común realizar tareas de cuidado personal, como maquillarse, mientras se está manejando, lo cual representa un peligro aún mayor mientras se está manejando.

Las conversaciones con los pasajeros del vehículo están muy normalizadas, y, sin embargo, resultan más peligrosas de lo que parecen, el 77% de los conductores se distrae de manera reiterada cuando habla al mismo tiempo que conduce (Vendrell, 2021). Si la conversación que se produce posee una carga emocional importante, estas distracciones aumentan. Es por ello que resulta importante evitar estas conversaciones en la medida de lo posible.

Derivado del análisis anteriormente presentado, se determinaron las siguientes distracciones para detectar en el presente proyecto.

- Enviando SMS con mano derecha
- Llamada con mano derecha
- Enviando SMS con mano izquierda
- Llamada con mano izquierda
- Manipulando la radio/GPS
- Ingeriendo bebidas
- Buscando algo en la parte de atrás
- Maquillando/Arreglando el cabello
- Hablando con pasajero

4.3 Etapa 3: Análisis de las bases de datos

Los problemas de aprendizaje automático comienzan con el conjunto de datos a utilizar, la elección de los datos y la cantidad de los mismos que es necesaria depende en gran parte del problema que se está tratando de investigar o resolver, elegir el conjunto de datos correcto es fundamental, pues si estos no son los adecuados puede conllevar a que se tenga un modelo incorrecto o menos preciso.

En la actualidad existen un gran número de bases de datos, las cuales contienen imágenes etiquetadas de conductores realizando distintas actividades. Las bases de datos son diferentes entre sí por la posición en la que se capturaron las imágenes, aportando diferente información de los conductores, además algunas de ellas varían de acuerdo al análisis de las distracciones que eligieron detectar. En la Ilustración 36 se muestran tres distintas perspectivas que adoptó la cámara para la

toma de fotografías de los conductores, en la imagen del inciso “a” se muestra la mitad superior de un conductor desde una vista frontal con un ángulo de inclinación, en la imagen del inciso “b” se observa la mitad superior de un conductor desde un punto de vista lateral, mientras que en el inciso “c” se muestra una imagen del rostro del conductor desde una perspectiva frontal.



Ilustración 36 Diferentes perspectivas de la cámara

Debido a la importancia y relación que existe entre la elección de la base de datos con encontrar el modelo óptimo para la solución que se quiere encontrar en cada problema es que se optó por utilizar bases de datos que contuvieran imágenes con una perspectiva similar a las de la Ilustración 36 inciso “a”, debido a que las distracciones que se van a analizar requieren la captura tanto del rostro del conductor como de sus brazos y manos.

La Tabla 2 presenta las características principales de la base de datos “Japanese Facial Expression”, mientras que la Ilustración 37 muestra un ejemplo del conjunto de imágenes de la base de datos, la cual fue analizada para su posible uso.

Tabla 2 Base de datos JAFFE

Base de datos	Descripción	Posición de la cámara	Distracciones
Japanese Facial Expression (JAFFE)	La base de datos contiene 213 imágenes de diez personas japonesas.	Frontal: solo se captura la cara de las personas.	Centrado en la detección de emociones: enojo, felicidad, neutro, tristeza, disgusto, miedo y sorpresa.



Ilustración 37 Imágenes de JAFFE

La base de datos “JAFFE” tuvo que ser descartada debido a que no cumplía con el requisito que debe de tener la perspectiva de la cámara al enfocar al conductor para poder analizar las distracciones que fueron planteadas con anterioridad.

La Tabla 3 detalla las características de la base de datos “Multimodal Multiview and Multispectral Driver Action Dataset”, de la misma manera la Ilustración 38 es un ejemplo de las imágenes que contiene la base de datos.

Tabla 3 Base de datos 3MDAD

Base de datos	Descripción	Posición de la cámara	Distracciones
<p>Multimodal Multiview and Multispectral Driver Action Dataset (3MDAD)</p>	<p>La base de datos contiene 100,000 imágenes de sujetos diferentes realizando 16 acciones distintas.</p>	<p>Frontal con inclinación: Se captura tanto el rostro y manos del conductor.</p>	<p>15 distracciones: Arreglarse el cabello, ajustar la radio, operar el GPS, escribir SMS, hablar por teléfono, ir tomando fotos, hablando con pasajero, cantando, fatiga, ingiriendo bebidas, buscando algo atrás, fumando.</p>



Ilustración 38 Imágenes de 3MDAD

La base de datos “3MDAD” cumplía con los requisitos de la perspectiva de la cámara y con el análisis de las distracciones deseadas pero fue descartada debido a que el conjunto de datos conlleva un análisis temporal de las distracciones, es decir, las acciones que realiza el conductor antes, durante y después de la distracción están etiquetadas como parte de la distracción, lo cual para el aprendizaje del modelo basado en redes neuronales convolucionales puede ocasionar una confusión en la clasificación de las clases debido a que las redes neuronales convolucionales no involucran un análisis temporal de las imágenes, se toma una decisión basado en a imagen capturada y se decide si se trata de una distracción o de un comportamiento normal del conductor.

La Tabla 4 muestra las características de la base de datos “Facial Expression Recognition”. La Ilustración 39 es un ejemplo de la base de datos y su contenido.

Tabla 4 Base de datos FER

Base de datos	Descripción	Posición de la cámara	Distracciones
Facial Expression Recognition (FER)	La base de datos contiene 15,000 imágenes de sujetos norteamericanos realizando expresiones.	Frontal: solo se captura la cara de las personas.	Centrado en la detección de emociones: enojo, felicidad, neutro, tristeza, disgusto, miedo y sorpresa.



Ilustración 39 Imágenes de FER

La base de datos “FER” tuvo que ser descartada para su uso de la misma manera en que fue descartada la base de datos “JAFPE”, debido a que se enfoca en el análisis de las emociones únicamente y la posición de la cámara con la que se capturaron las imágenes no es la ideal por solo capturar el rostro de las personas.

En la Tabla 5 se pueden observar las características de la base de datos “State Farm Distracted Driver Detection”. En la Ilustración 40 se muestran, como ejemplo, 4 imágenes de la base de datos.



Ilustración 40 Imágenes de SFDDD

Tabla 5 Base de datos SFDDD

Base de datos	Descripción	Posición de la cámara	Distracciones
State Farm Distracted Driver Detection (SFDDD)	La base de datos contiene 100,000 imágenes de conductores realizando 10 acciones diferentes.	Frontal con inclinación: se captura tanto el rostro, brazos y manos del conductor.	Conducción segura, enviando SMS, hablando por teléfono, operando la radio/GPS, ingiriendo bebidas, buscando algo atrás, arreglándose el cabello/maquillándose, hablando con pasajero.

Tras haber examinado las bases de datos presentadas, se decidió seleccionar la base de datos “*State Farm Distracted Driver Detection*” (SFDDD) (Kaggle, 2022) para la realización de un conjunto de pruebas, con la finalidad de poder comprender el comportamiento de la detección de distracciones y poder comparar los resultados obtenidos entre modelos entrenados desde cero (modelos desarrollados por el programador) y los modelos preentrenados (transferencia de aprendizaje).

El conjunto de datos “*State Farm Distracted Driver Detection*” se caracteriza de las demás bases de datos porque cumple con las dos características principales

mencionadas anteriormente en la definición de distracciones del proyecto, como primera característica las imágenes fueron tomadas desde una posición adecuada de la cámara (frontal con una inclinación), de esta manera es posible evaluar la mitad superior del cuerpo del conductor, involucrando rostro y brazos del mismo para su análisis, la segunda característica es que la base de datos contiene las distracciones planteadas con anterioridad, aumentando la clase de “conducción segura” y la evaluación del uso del celular para enviar mensajes y realizar llamadas tanto con la mano derecha como con la mano izquierda.

La incorporación de la clase “conducción segura” representa un gran aporte para el funcionamiento del sistema debido a que la conducción segura es el estado ideal el cual deben de ejecutar todos los conductores normalmente, así mismo, le será de utilidad al sistema para poder identificar la variación entre una conducción normal y una distracción, además de que el sistema no reportará ninguna recomendación mientras se esté llevando a cabo una buena conducción y genere alertas únicamente cuando detecte alguna de las distracciones a clasificar.

La base de datos contiene 100,000 imágenes etiquetadas, 10,000 imágenes por cada clase, para la identificación de 10 clases, las cuales se enlistan en la Tabla 6. En la columna de la izquierda se tiene la clave con la cual fueron etiquetadas las imágenes, mientras que la columna de la derecha indica el nombre de la distracción que representa dicha clave.

Tabla 6 Clases del Dataset

Clave	Nombre
C0	Conducción segura
C1	Enviando SMS con mano derecha
C2	Llamada con mano derecha
C3	Enviando SMS con mano izquierda
C4	Llamada con mano izquierda
C5	Manipulando la radio/GPS
C6	Ingiriendo bebidas
C7	Buscando algo en la parte de atrás
C8	Maquillando/Arreglando el cabello
C9	Hablando con pasajero

En la Ilustración 41 se puede observar una muestra de las diferentes imágenes para cada clase encontradas en esta base de datos.



Ilustración 41 Imágenes de la base de datos

4.3.1 Protocolo de realización de pruebas con base de datos “State Farm Distracted Driver Detection”

Para el desarrollo del proyecto se optó por realizar un modelo de aprendizaje basado en las redes neuronales artificiales convolucionales debido a que este tipo de modelos son los que mejor se adecúan a los problemas de clasificación multiclase, además se adecúa muy bien al tratamiento con imágenes, obteniendo buenas predicciones en sus resultados. Para poder ejecutar modelos de redes neuronales convolucionales es necesario llevar a cabo un protocolo para realizar múltiples entrenamientos en la búsqueda de un modelo óptimo, el protocolo que se propone a seguir durante los entrenamientos de este proyecto se detalla en la Ilustración 42.

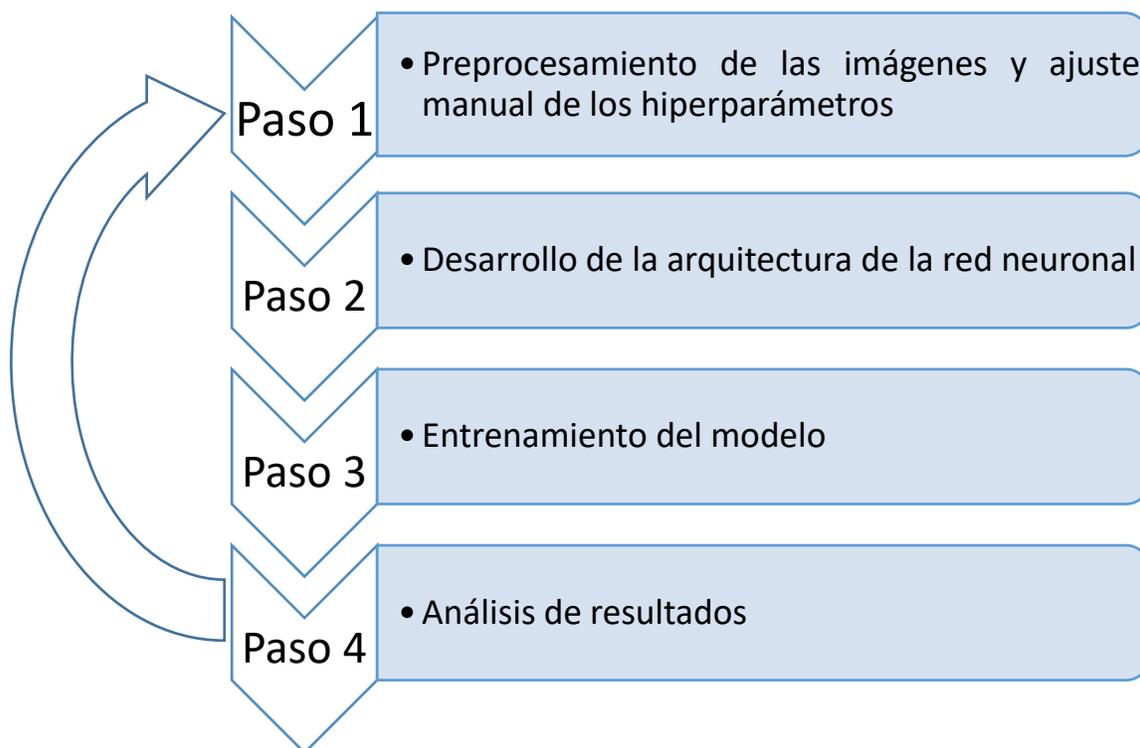


Ilustración 42 Protocolo de entrenamiento para los modelos

El protocolo propuesto incluye en el primer paso una selección manual de los hiperparámetros (tasa de aprendizaje, tamaño del lote y número de épocas) a utilizar durante el entrenamiento, estos parámetros son ajustables por el programador, es común que el proceso sea tedioso y costoso por las pruebas encargadas de buscar el modelo óptimo, sin embargo, algunos autores como Simon (2009) recomiendan iniciar los entrenamientos con valores pequeños, para el caso de la tasa de aprendizaje se recomiendan valores entre 0 y 1, mayormente como punto de partida el 0.001 y dependiendo de los resultados obtenidos por el modelo es que se modifican estos mismos hiperparámetros. Para el caso del tamaño del lote y el número de épocas se suelen usar valores mayores a uno, con la particularidad que el tamaño del lote puede ser igual o menor que el número de muestras del conjunto de datos.

En el paso 2 se desarrolla la arquitectura del modelo de la red neuronal, es decir, se elige el número de capas convolucionales, el número de capas totalmente conectadas y el uso de herramientas como el *“Max-pooling”* y el *“dropout”* con la finalidad de obtener las mejores características de las imágenes y evitar problemas comunes en los entrenamientos como el *“overfitting”*.

En el paso 3 se ejecuta el modelo de aprendizaje previamente desarrollado para la clasificación de las imágenes, en donde dependiendo de la profundidad de las capas ocultas y de los parámetros entrenables, corresponderá al tiempo que tardarán los entrenamientos.

En el paso 4 se analizan los datos y gráficas arrojadas por la red, tales como el valor de exactitud alcanzado por el entrenamiento, las gráficas de exactitud y pérdida

para observar el comportamiento del entrenamiento durante cada época y saber si disminuir o incrementar alguno de los hiperparámetros para poder mejorar el modelo. Si los resultados no son los esperados se procede a volver a repetir el protocolo de entrenamiento hasta poder obtener mejores resultados.

4.3.2 Preprocesamiento de las imágenes

Si bien las redes neuronales convolucionales desempeñan un buen trabajo en la clasificación de imágenes, estos mismos algoritmos pueden volverse ineficientes cuando el tamaño de las imágenes es muy grande debido a que a mayor tamaño de la imagen mayor será el número de píxeles que producen un aumento exponencial de las variables de entrada del algoritmo, imposibles de manejar para una red neuronal artificial de arquitectura tradicional. Lo anterior conlleva un aumento en el costo computacional, por lo tanto, una manera de reducir este problema es redimensionar las imágenes para obtener el menor tamaño posible, pero sin llegar a perder las características principales de las imágenes, este proceso se lleva a cabo antes de diseñar los modelos y ejecutar los entrenamientos.

El redimensionamiento aplicado a las imágenes lleva a cabo una reducción del total del número de píxeles que contiene cada una de las imágenes, con la particularidad de seguir conservando los detalles de la misma, es decir no se está realizando ningún recorte dentro de la imagen que pueda quitar información, aunque en desventaja un redimensionamiento abrupto de igual forma podría conllevar a una pérdida de información, este nuevo tamaño de redimensionamiento deseado es

proporcionado por el programador, es por ello que se debe tener cuidado con no afectar la información contenida dentro de las imágenes al llevar a cabo este proceso.

Para el conjunto de datos “*State Farm Distracted Driver Detection*” que cuenta con imágenes de unas dimensiones de 640x480 (imágenes grandes que conllevan un alza en el tiempo de entrenamiento) fue necesario aplicar un redimensionamiento de 48x48 (Ilustración 43) el menor tamaño con el cual, para esta base de datos, se reduce el costo computacional de los entrenamientos y a su vez no se pierde/deforma la información contenida en las imágenes de la base de datos.



Ilustración 43 Ajuste de las imágenes

Como paso posterior se agruparon los datos en conjuntos de entrenamiento, validación y test, esto con la finalidad de asegurarnos que el modelo aprenda correctamente con imágenes distintas y evitar así que memorice o clasifique de una manera errónea.

Una vez teniendo listos las imágenes divididas en conjuntos de entrenamiento y validación se procedió a realizar la conversión de las imágenes originales en RGB (3 canales) a escala de grises, las imágenes a escala de grises (1 canal) simplifican los cálculos matemáticos y por ende este proceso se convierte en otra forma de poder reducir los recursos computacionales en cada entrenamiento, reduciendo así el tiempo del mismo. Los pasos anteriores son aplicados antes de iniciar cada entrenamiento por lo que los modelos de aprendizaje trabajaron con las imágenes obtenidas después de realizar el anterior preprocesamiento de las imágenes.

4.3.3 Desarrollo de los modelos y entrenamientos desde cero

La arquitectura de los modelos presentados a continuación fueron desarrollados desde cero, en donde a base de entrenamientos, se le fueron incorporando o quitando capas convolucionales y totalmente conectadas a los modelos, de la misma forma se hizo uso u omisión de herramientas de extracción de características.

En la Tabla 7 se encuentran los hiperparámetros utilizados para los primeros cuatro modelos de entrenamiento, estos hiperparámetros se seleccionaron basado en la propuesta de Simon (2009).

De la misma forma, la Tabla 8 contiene las características de la arquitectura utilizada para el primer modelo de entrenamiento, en la cual, en primera instancia se logra apreciar que entre cada capa convolucional se agregaron dos capas más, una capa de “*Max-Pooling*” y otra de “*dropout*”, como menciona Aggarwal (2018), estos

filtros dependiendo de la naturaleza de los datos y de la problemática, pueden ser de gran utilidad para disminuir el tiempo de entrenamiento y poder obtener un buen desempeño en el entrenamiento pero al igual que pasa con los hiperparámetros, son ajustados por el programador, en total se cuenta con 5 capas convolucionales, 5 capas “*Max-pooling*”, 7 capas “*dropout*” y 3 capas totalmente conectadas. El primer modelo alcanzó una exactitud del 75% por ello se decidió modificar la arquitectura de la red neuronal. Como menciona Chollet (2017), la búsqueda de un modelo que se adapte a un problema de clasificación requiere la realización de entrenamientos y la manipulación de los parámetros de la red, algunas veces puede ser de utilidad una mayor profundidad en las capas o el uso de filtros de extracción de características.

Para el segundo modelo (Tabla 9) se optó por implementar dos capas convolucionales seguidas, antes de los filtros de “*Max-pooling*” y “*dropout*”, este nuevo modelo cuenta con 6 capas convoluciones, 3 capas “*Max-pooling*”, 3 capas “*dropout*” y una capa totalmente conectada, obteniendo una exactitud del 65% por lo que fue necesario probar con un nuevo modelo.

En el tercer modelo (Tabla 10) se trabajó con 7 capas convolucionales, 4 capas “*Max-Pooling*”, 2 capas de “*dropout*” y 2 capas totalmente conectadas, este nuevo modelo consiguió mejorar el resultado de los dos modelos anteriores alcanzando un 83% en exactitud.

Se realizó una última modificación esperando poder mejorar el resultado del modelo anterior, en el modelo cuatro (Tabla 11) se utilizaron 5 capas convolucionales,

5 capas “*Max-Pooling*”, 5 capas “*dropout*” y 3 capas totalmente conectadas, logrando alcanzar una exactitud del 90%.

Derivado de las experimentaciones con los primeros cuatro modelos se pudo observar que el uso del filtro “*Max-pooling*” ayuda en gran medida a poder rescatar las características más sobresalientes después de aplicar los filtros convolucionales, por otra parte, las capas de “*dropout*” son igual de importantes pero hay que experimentar el porcentaje que debemos darle en cada uso, debido a que se trata del “apagado” de neuronas no es muy recomendable abusar de su uso, Goodfellow, Bengio y Courville (2016) recomiendan aplicar un valor de “*dropout*” del 0.2 en las primeras capas de entrada e ir incrementando su valor en las capas profundas hasta un 0.5 máximo, como mencionan un valor muy pequeño puede no generar cambios y uno muy elevado podría afectar severamente al aprendizaje del modelo.

Tabla 7 Hiperparámetros de los primeros 4 modelos

Hiperparámetros de los modelos 1 al 4	
Tasa de aprendizaje	0.001
Tamaño del lote	150
Número de épocas	15

Tabla 8 Arquitectura de la red del modelo 1

32 CONV 2D
MaxPooling2D
Dropout (0.3)
64 CONV 2D
MaxPooling2D
Dropout (0.2)
128 CONV 2D
MaxPooling2D
Dropout (0.3)
256 CONV 2D
MaxPooling2D
Dropout (0.2)
512 CONV 2D
MaxPooling 2D
Dropout (0.5)
Flatten
Dense 128 units
Dropout (0.2)
Dense 256 units
Dropout (0.2)
Dense 10 units

En el segundo modelo (Tabla 9) se experimentó eliminando capas totalmente conectadas al final.

Tabla 9 Arquitectura de la red modelo 2

32 CONV 2D
32 CONV 2D
MaxPooling2D
Dropout (0.2)
64 CONV 2D
64 CONV 2D
MaxPooling2D
Dropout (0.3)
128 CONV 2D
128 CONV 2D
MaxPooling2D
Dropout (0.4)
Flatten
Dense 10 units

Para el modelo de entrenamiento 3 (Tabla 10) se probó realizar una arquitectura más profunda a las anteriores al incrementa el número de capas convolucionales y restringir en gran medida el uso del “*dropout*”.

Tabla 10 Arquitectura de la red modelo 3

32 CONV 2D
MaxPooling2D
64 CONV 2D
64 CONV 2D
MaxPooling2D
Dropout (0.2)
128 CONV 2D
128 CONV 2D
MaxPooling2D
256 CONV 2D
256 CONV2D
MaxPooling2D
Dropout (0.5)
Flatten
Dense 128 units
Dense 10 units

Para el modelo de entrenamiento 4 (Tabla 11) se buscó implementar un modelo con el mismo número de capas convolucionales y el mismo número de filtros de extracción de características y de “*dropout*”.

Tabla 11 Arquitectura de la red modelo 4

32 CONV 2D
MaxPooling2D
Dropout (0.3)
64 CONV 2D
MaxPooling2D
128 CONV 2D
MaxPooling2D
Dropout (0.2)
256 CONV 2D
MaxPooling2D
Dropout (0.3)
512 CONV 2D
MaxPooling2D
Dropout (0.5)
Flatten
Dense 128 units
Dense 256 units
Dropout (0.2)
Dense 10 units

Las primeras 4 arquitecturas de redes neuronales convolucionales compartieron la misma tasa de aprendizaje, el mismo tamaño del lote y el mismo número de épocas, pero a pesar de ello cada uno de los modelos obtuvo diferentes resultados, esto debido

a que cada modelo se diferenciaba uno del otro ya sea por la profundidad de la red neuronal, el uso de técnicas que ayudan a encontrar las características más sobresalientes de las imágenes y el uso del “*dropout*”. A modo de comparación se observan los resultados obtenidos por los primeros 4 modelos de entrenamiento en la Tabla 12, en donde se observa que el modelo 4 es el que obtuvo mejores resultados.

Tabla 12 Resultados de las pruebas de los modelos 1 al 4

	Modelo 1	Modelo 2	Modelo 3	Modelo 4
Accuracy	0.75	0.65	0.83	0.90
Loss	0.49	1.35	1.09	0.76
Val Accuracy	0.79	0.73	0.86	0.93
Val Loss	0.53	1.87	0.89	0.63

Para los siguientes 4 modelos de entrenamiento se probó la opción de variar los hiperparámetros anteriormente utilizados, tratando de mejorar los resultados obtenidos por el modelo 4, los nuevos hiperparámetros son mostrados en la Tabla 13, siguiendo los consejos de Haykin S. (2009) se optó por incrementar el valor de cada uno de ellos.

Tabla 13 Variación de los hiperparámetros

Hiperparámetros de los modelos 5 al 8	
Tasa de aprendizaje	0.003
Tamaño del lote	200
Número de épocas	30

Lo modelos de entrenamiento 5, 6, 7 y 8 son los mismos modelos de las arquitecturas 1, 2, 3 y 4 respectivamente, la finalidad de estos entrenamientos es analizar como influyen los hiperparámetros y sus modificaciones con los resultados.

A modo de comparación se muestran los resultados de los modelos del 5 al 8 presentados en la Tabla 14, en donde se observa que el modelo 8 es el que obtuvo mejores resultados dentro de los 4 últimos experimentos, no obstante, comparando los resultados de la Tabla 12 con la Tabla 18, se puede apreciar que el modelo 4 fue el que obtuvo el mejor resultado de precisión de los 8 modelos, con un total de 90% de precisión en el entrenamiento y 93% en validación.

Tabla 14 Resultados de las pruebas

	Modelo 5	Modelo 6	Modelo 7	Modelo 8
Exactitud	0.72	0.57	0.75	0.88
Pérdida	0.89	1.58	1.09	0.96
Exactitud Val.	0.78	0.59	0.80	0.90
Pérdida Val.	1.23	1.97	0.89	0.83

Cabe destacar que un pequeño cambio tanto en la arquitectura en los modelos de entrenamiento como en los hiperparámetros pueden provocar grandes cambios en los resultados de aprendizaje de la red, la ventaja con la que cuenta la base de datos

“*State Farm Distracted Driver Detection*” es que se cuenta con una buena cantidad de datos para probar entrenamientos desde cero, mientras que la desventaja recae en que son demasiadas imágenes las cuales tienen que ser procesadas en cada modelo de entrenamiento, lo que representa un elevado costo computacional.

4.3.4 Transferencia de aprendizaje

Se decidió explorar la técnica “transferencia de aprendizaje” para la base de datos “*State Farm Distracted Driver Detection*” para lo cual se realizaron 2 entrenamientos, el primero utilizando la red preentrenada “MobileNet-v2” y el segundo usando la red “VGG-16”.

La red “MobileNet-v2” es una red neuronal convolucional, la cual cuenta con 53 capas de profundidad, la red fue entrenada para poder clasificar imágenes en 1000 clases diferentes de objetos, el tamaño de entrada de las imágenes en la red es de 224 por 224 y su arquitectura se muestra en la Ilustración 44 (Sandler, Howard, Zhu, Zhmoginov y Chen, 2018).

La red “VGG-16” es una red neuronal convolucional con 16 capas de profundidad y al igual que la red “MobileNet-v2” puede clasificar imágenes en 1000 clases diferentes, el tamaño de entrada de las imágenes es de 224 por 224, la arquitectura de la red se muestra en la Ilustración 45 (MATLAB, 2023).

Type	Filter Shape	Input Size
Conv1	3×3×3×32	224×224×3
Conv2 dw	3×3×32 dw	112×112×32
Conv2 pw	1×1×32×64	112×112×32
Conv3 dw	3×3×64 dw	112×112×64
Conv3 pw	1×1×64×128	56×56×64
Conv4 dw	3×3×128 dw	56×56×128
Conv4 pw	1×1×128×128	56×56×128
Conv5 dw	3×3×128 dw	56×56×128
Conv5 pw	1×1×128×256	28×28×128
Conv6 dw	3×3×256 dw	28×28×256
Conv6 pw	1×1×256×256	28×28×256
Conv7 dw	3×3×256 dw	28×28×256
Conv7 pw	1×1×256×512	14×14×256
Conv8 dw	3×3×512 dw	14×14×512
Conv8 pw	1×1×512×512	14×14×512
Conv9 dw	3×3×512 dw	14×14×512
Conv9 pw	1×1×512×512	14×14×512
Conv10 dw	3×3×512 dw	14×14×512
Conv10 pw	1×1×512×512	14×14×512
Conv11 dw	3×3×512 dw	14×14×512
Conv11 pw	1×1×512×512	14×14×512
Conv12 dw	3×3×512 dw	14×14×512
Conv12 pw	1×1×512×512	14×14×512
Conv13 dw	3×3×512 dw	14×14×512
Conv13 pw	1×1×512×1024	7×7×512
Conv14 dw	3×3×1024 dw	7×7×1024
Conv14 pw	1×1×1024×1024	7×7×1024
Avg Pool	Pool 7×7	7×7×1024
FC	1024×25	1×1×1024
Softmax	Classifier	1×1×25

Ilustración 44 Arquitectura "MobileNet-v2" obtenido de (Sandler, Howard, Zhu, Zhmoginov y Chen, 2018).

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 224, 224, 64)	1792
conv2d_2 (Conv2D)	(None, 224, 224, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 112, 112, 64)	0
conv2d_3 (Conv2D)	(None, 112, 112, 128)	73856
conv2d_4 (Conv2D)	(None, 112, 112, 128)	147584
max_pooling2d_2 (MaxPooling2D)	(None, 56, 56, 128)	0
conv2d_5 (Conv2D)	(None, 56, 56, 256)	295168
conv2d_6 (Conv2D)	(None, 56, 56, 256)	590080
conv2d_7 (Conv2D)	(None, 56, 56, 256)	590080
max_pooling2d_3 (MaxPooling2D)	(None, 28, 28, 256)	0
conv2d_8 (Conv2D)	(None, 28, 28, 512)	1180160
conv2d_9 (Conv2D)	(None, 28, 28, 512)	2359808
conv2d_10 (Conv2D)	(None, 28, 28, 512)	2359808
max_pooling2d_4 (MaxPooling2D)	(None, 14, 14, 512)	0
conv2d_11 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_12 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_13 (Conv2D)	(None, 14, 14, 512)	2359808
max_pooling2d_5 (MaxPooling2D)	(None, 7, 7, 512)	0
flatten_1 (Flatten)	(None, 25088)	0
dense_1 (Dense)	(None, 4096)	102764544
dropout_1 (Dropout)	(None, 4096)	0
dense_2 (Dense)	(None, 4096)	16781312
dropout_2 (Dropout)	(None, 4096)	0
dense_3 (Dense)	(None, 2)	8194
Total params: 134,268,738		
Trainable params: 134,268,738		
Non-trainable params: 0		

Ilustración 45 Arquitectura "VGG-16" obtenido de (MATLAB, 2023).

Los resultados obtenidos por cada una de las redes preentrenadas se puede apreciar en la Tabla 19.

Tabla 15 Respuesta de redes preentrenadas

Resultados	MobileNet-v2	VGG-16
Exactitud	0.93	0.92
Pérdida	0.34	0.45

Los modelos “MobileNet-v2” y “VGG-16” obtuvieron buenos resultados, sin embargo, ambos modelos son muy pesados debido a la gran cantidad de parámetros y pesos que implican, debido a esto y aunado a que la base de datos “*State Farm Distracted Driver Detection*” cuenta con 100,000 imágenes, el proceso de entrenamiento se volvió costoso, implicando un mayor tiempo de entrenamiento y resultando en un modelo pesado para la aplicación móvil.

4.3.5 Modelo óptimo entrenado desde cero

Derivado de los entrenamientos anteriores se optó por realizar una prueba más con los modelos desarrollados y entrenados desde cero, anteriormente se había propuesto una serie de hiperparámetros utilizados en la literatura, posteriormente se probó incrementando estos valores, en esta ocasión se propuso disminuir el valor de los hiperparámetros de entrenamiento, los cuales se observan en la Tabla 20.

Tabla 16 Nuevos hiperparámetros

Hiperparámetros	
Tasa de aprendizaje	0.0001
Tamaño del lote	200
Número de épocas	25

La nueva arquitectura de la red neuronal convolucional consta de 5 capas convolucionales, 3 capas totalmente conectadas, 5 capas de filtros “*Max-pooling*” y 7 capas “*dropout*”, en esta ocasión se utilizaron capas “*dropout*” tanto en las capas de entrada de la red, las capas ocultas y en las capas de salida (capas totalmente conectadas), como menciona Ekman (2021) dependiendo del tipo de problema que se enfrenta, en algunas ocasiones se recomienda el uso del “*dropout*” tanto en las capas iniciales como en las capas de salida de la red, en las cuales se llevan a cabo más procesos computacionales para dar paso a la clasificación de las imágenes, también recomienda que el “*dropout*” vaya acompañado de una capa de “*Max-pooling*” tal y como se venía haciendo con anterioridad.

La arquitectura de la nueva red convolucional a la que se le aplicó la nueva configuración de hiperparámetros se presenta en la Ilustración 46.

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 48, 48, 32)	320
batch_normalization (Batch Normalization)	(None, 48, 48, 32)	128
activation (Activation)	(None, 48, 48, 32)	0
max_pooling2d (MaxPooling2D)	(None, 24, 24, 32)	0
dropout (Dropout)	(None, 24, 24, 32)	0
conv2d_1 (Conv2D)	(None, 24, 24, 64)	51264
batch_normalization_1 (Batch Normalization)	(None, 24, 24, 64)	256
activation_1 (Activation)	(None, 24, 24, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 12, 12, 64)	0
dropout_1 (Dropout)	(None, 12, 12, 64)	0
conv2d_2 (Conv2D)	(None, 12, 12, 128)	73856
batch_normalization_2 (Batch Normalization)	(None, 12, 12, 128)	512
activation_2 (Activation)	(None, 12, 12, 128)	0
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 128)	0
dropout_2 (Dropout)	(None, 6, 6, 128)	0
conv2d_3 (Conv2D)	(None, 6, 6, 256)	295168
batch_normalization_3 (Batch Normalization)	(None, 6, 6, 256)	1024
activation_3 (Activation)	(None, 6, 6, 256)	0
max_pooling2d_3 (MaxPooling2D)	(None, 3, 3, 256)	0
dropout_3 (Dropout)	(None, 3, 3, 256)	0
conv2d_4 (Conv2D)	(None, 3, 3, 512)	1180160
batch_normalization_4 (Batch Normalization)	(None, 3, 3, 512)	2048
activation_4 (Activation)	(None, 3, 3, 512)	0
max_pooling2d_4 (MaxPooling2D)	(None, 1, 1, 512)	0
dropout_4 (Dropout)	(None, 1, 1, 512)	0
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 128)	65664
batch_normalization_5 (Batch Normalization)	(None, 128)	512
activation_5 (Activation)	(None, 128)	0
dropout_5 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 256)	33024
batch_normalization_6 (Batch Normalization)	(None, 256)	1024
activation_6 (Activation)	(None, 256)	0
dropout_6 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 10)	2570
Total params: 1,707,530		
Trainable params: 1,704,778		
Non-trainable params: 2,752		

Ilustración 46 Modelo óptimo

Con la selección de los hiperparámetros vista en la Tabla 20 se realizó el entrenamiento del nuevo modelo y se obtuvieron los resultados contenidos en la Tabla 21, en la que se puede observar que el modelo alcanzó una exactitud del 93% en entrenamiento y del 97% en validación.

Tabla 17 Resultados del modelo

Entrenamiento	Exactitud	0.93
	Pérdida	0.21
Validación	Exactitud	0.97
	Pérdida	0.095

4.3.6 Evaluación del modelo con mejores resultados

Después de haber entrenado distintos modelos se tomó aquel que obtuvo el mejor resultado en exactitud, para lo cual se procede a generar y evaluar las gráficas de exactitud y pérdida del modelo para confirmar que se obtuvo un buen entrenamiento. A partir de la Ilustración 47 se puede apreciar un buen desempeño del entrenamiento dado por la gráfica de exactitud.

Si las curvas de exactitud tanto del entrenamiento como de la validación son cercanas entre sí, significa que el modelo entrenado está generalizando bien y es viable el poder utilizarlo. Por el contrario, si la exactitud en validación es muy distinta

al de entrenamiento tanto por encima como por debajo, entonces esto es un indicador de que nuestro modelo no se ha entrenado bien y no es viable utilizarlo, esto también es un posible indicador de “*overfitting*”, lo que significa que el modelo de aprendizaje solo se ajusta en detectar casos particulares que le enseñemos y será incapaz de reconocer nuevos datos de entrada.

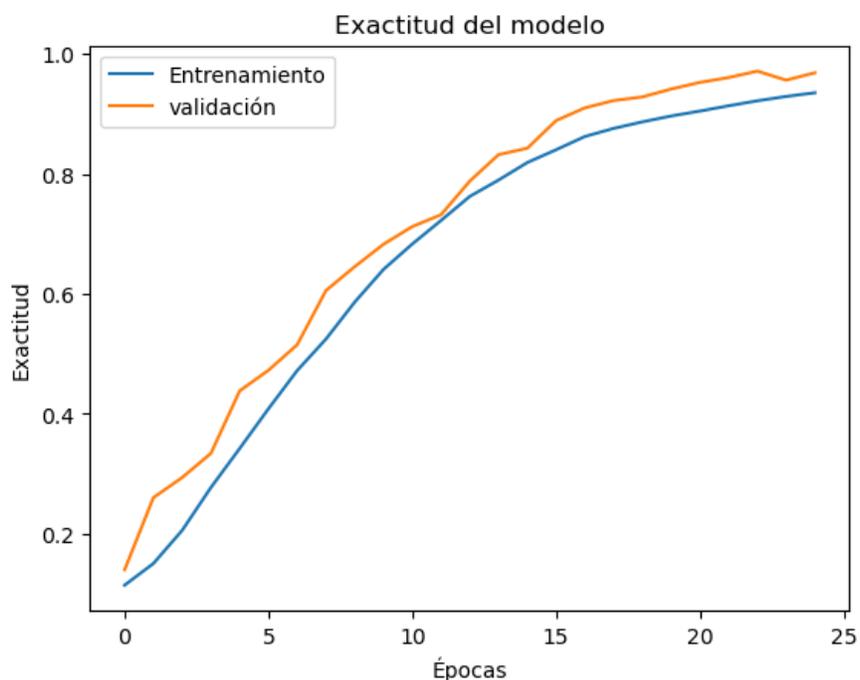


Ilustración 47 Gráfica de exactitud

La grafica de pérdida o “*Loss*” en inglés (Ilustración 48) indica el error entre la salida del modelo frente a la salida que deseamos, es decir, la pérdida es un valor que representa la suma total de errores en los modelos, mide que tan bien o mal está funcionando, si los errores son altos, la pérdida será alta, lo que significa que el modelo

no hace un buen trabajo, y en caso contrario, cuanto menor sea la pérdida mejor funcionará el modelo.

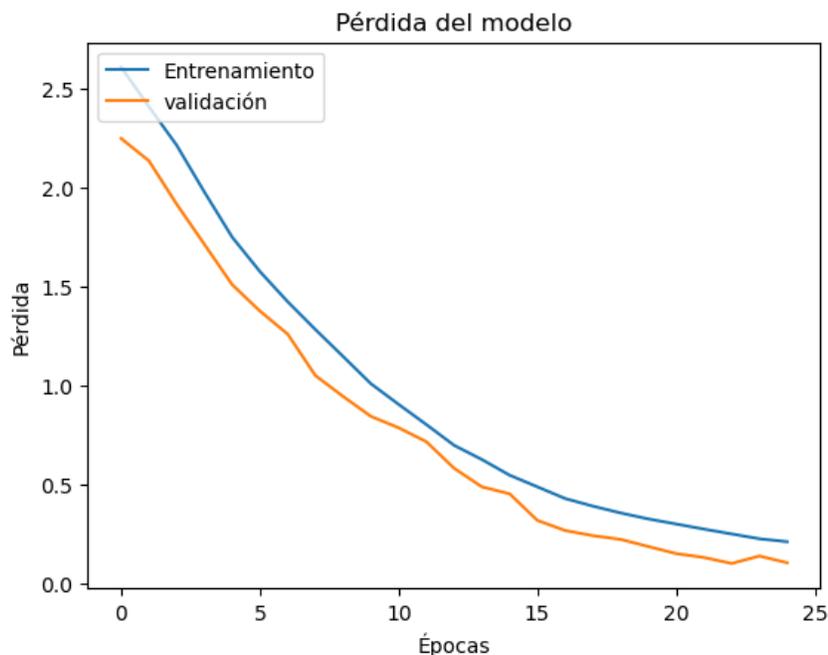


Ilustración 48 Gráfica de pérdida

Por último, para poder observar más a detalle el funcionamiento del modelo se construyó su “matriz de confusión para la clasificación multiclase” (Ilustración 49), la cual es una tabla en la que se muestra cada clase de los datos de evaluación y el número de predicciones correctas e incorrectas, teniendo en la diagonal de izquierda a derecha las predicciones acertadas de la red, mientras que las demás casillas muestran el número de veces en las que falló el modelo de predicción.

En el eje de las “Y” de la Ilustración 49 se encuentra la respuesta deseada, en donde los números del 0 al 9 representan a cada una de las clases en el orden en el

que se encuentran en la Tabla 22. Mientras que en el eje de las “X” encontramos la respuesta obtenida por el modelo.

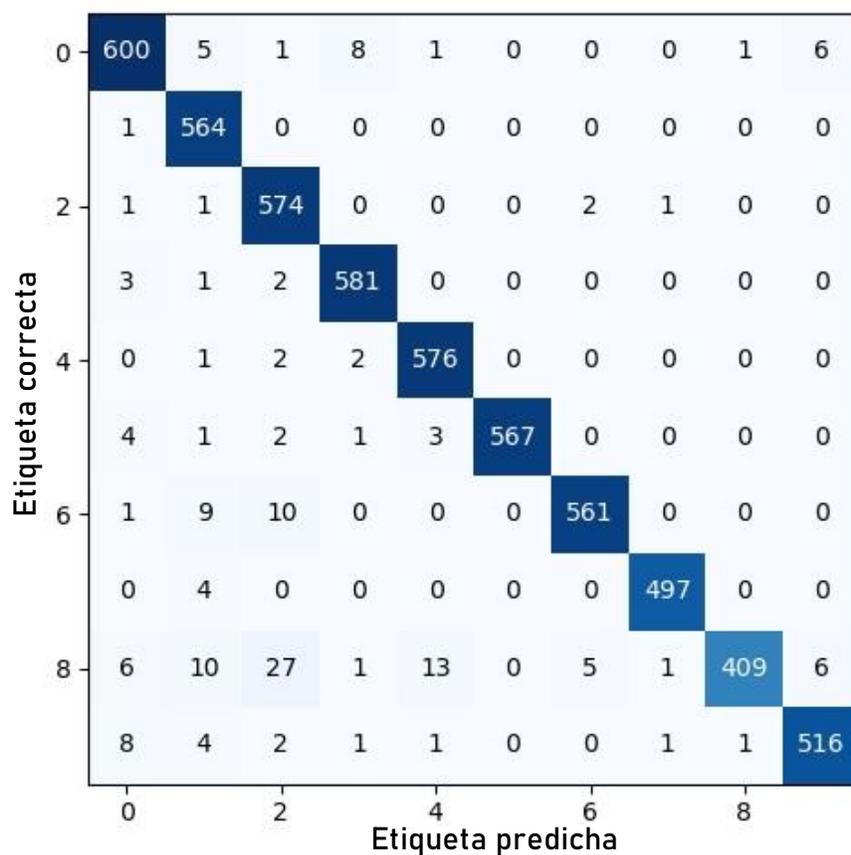


Ilustración 49 Matriz de confusión del modelo

De esta manera se puede apreciar que se obtuvieron buenos resultados en la clasificación, teniendo como la clase mejor clasificada a la clase 0 la cual corresponde a la etiqueta “Conducción segura” con un total de 600 aciertos de 622 en el test.

Y la clase que más trabajo le costó al modelo clasificar y que contiene más errores es la clase 8 la cual corresponde a la etiqueta “Maquillando/Arreglando el cabello” con un total de 69 errores de 478 datos evaluados en el test.

Tabla 18 Representación de las etiquetas de la matriz de confusión

Número	Clase a la que corresponde
0	Conducción segura
1	Enviando SMS con mano derecha
2	Llamada con mano derecha
3	Enviando SMS con mano izquierda
4	Llamada con mano izquierda
5	Manipulando la radio/GPS
6	Ingiriendo bebidas
7	Buscando algo en la parte de atrás
8	Maquillando/Arreglando el cabello
9	Hablando con pasajero

4.4 Etapa 4: Protocolo de captura de imágenes y generación de la base de datos local

Para la generación de la base de datos se propuso el siguiente protocolo de captura de imágenes detallado en la Ilustración 50.

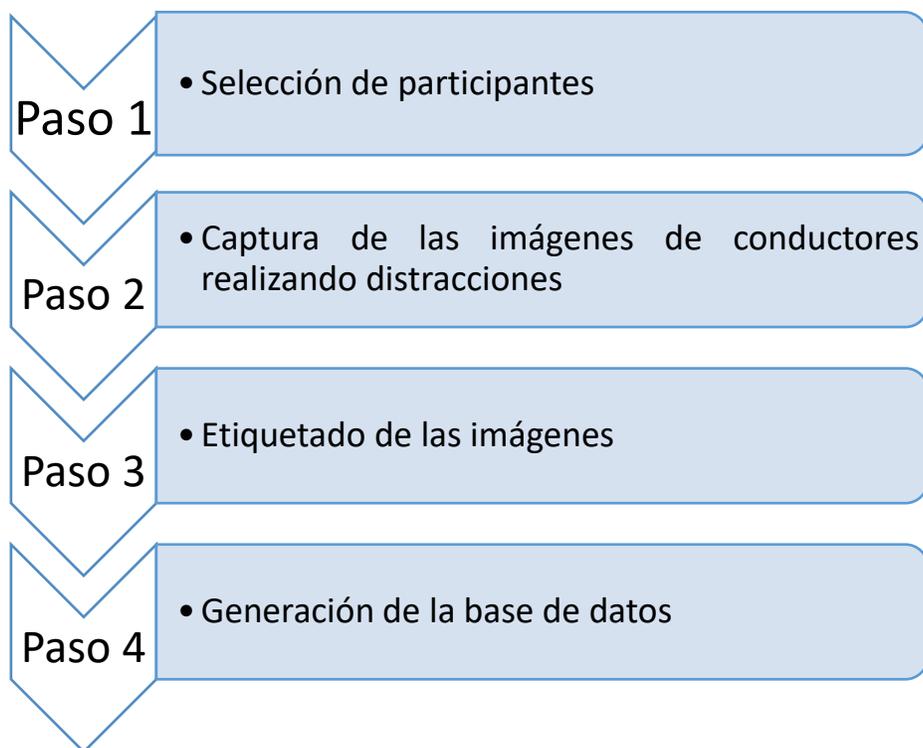


Ilustración 50 Protocolo de captura de imágenes y generación de la base de datos

Para la construcción de la base de datos se siguió una serie de 4 pasos importantes, pues al igual que durante el análisis de las diferentes bases de datos, es indispensable contar con una buena selección y captura de imágenes que sean de utilidad para los entrenamientos de las redes neuronales convolucionales. Se contempló la toma de fotografías de los conductores representando las 10 clases anteriormente seleccionadas y encontradas también en la base de datos “*State Farm Distracted Driver Detection*”.

El primer paso constó en delimitar la muestra de los datos en donde se establecieron las características principales con las que debían de contar los participantes de la muestra.

En el segundo paso se llevó a cabo la captura de las imágenes mediante el un dispositivo móvil, el cual capturaba imágenes de las acciones de los participantes cada segundo.

Durante el tercer paso se llevó a cabo un preprocesamiento en las imágenes para realizar su posterior etiquetado

En el cuarto paso se realizó la división de los datos para asignarlos con su respectiva clase.

4.4.1 Paso 1: Selección de participantes

Los participantes seleccionados para la captura de las imágenes fueron conductores de automóviles de entre 18 y 40 años de edad, de ambos sexos, que contaran con una licencia de conducir vigente. Si bien en México no hay una edad máxima para manejar se tomó en cuenta que por lo general los síntomas de vista cansada y pérdida de la misma comienzan a presentarse entre los 40 y 45 años de edad (Clínica Rahhal oftalmología, 2023), es por ello que no se tomaron en cuenta personas que superarán este rango de edad, así mismo, en nuestro país se considera que una persona cumple su mayoría de edad a los 18 años, siendo capaz de tomar una mayor responsabilidad de sus acciones y los riesgos que estos conllevan, como en el caso de la conducción, por estos motivos se seleccionó este rango de edad para la generación de la base de datos.

Se contó con un total de 32 participantes a los cuales se les dio indicaciones de realizar las actividades de las 10 clases correspondientes a predecir por el sistema, dichas actividades se llevaron a cabo en un ambiente controlado debido al peligro que conlleva la conducción distraída, se trabajó con distintos vehículos puestos en marcha a una baja velocidad sobre avenidas no transitadas, la conducción se realizó durante las condiciones de la luz del día.

4.4.2 Paso 2: Captura de las imágenes

Para la toma de las fotografías fue necesario contar con la cámara de un dispositivo móvil y con un soporte para celulares adaptable para su uso en automóviles como el que se muestra en la Ilustración 51.



Ilustración 51 Soporte para celular

El soporte para dispositivos móviles fue colocado en dos lugares que capturaban la misma perspectiva del conductor dentro de los vehículos, ambas posiciones no afectaban la perspectiva de las imágenes, únicamente eran para la comodidad de cada uno de los conductores, a la mitad de los participantes se les colocó el soporte en el tablero del vehículo y a la otra mitad se les colocó en el parabrisas.

Se configuró el dispositivo móvil mediante una aplicación para indicarle a la cámara que capturara fotografías del conductor cada segundo, con la finalidad de obtener la mayor información posible.

4.4.3 Paso 3: Etiquetado de las imágenes

Una vez fueron capturadas las imágenes de los conductores realizando las actividades de las clases a clasificar fue necesario descartar todas aquellas imágenes que no contuvieran información importante para desarrollar los entrenamientos de los modelos (i.e., durante el cambio de cada actividad a realizar).

La base de datos generada terminó con un total de 10,000 imágenes, 1,000 por cada una de las 10 clases anteriormente seleccionadas, sin embargo, las imágenes capturadas por las cámaras de los dispositivos móviles tienden a ser grandes y pesadas, cada una de ellas tenía las dimensiones de 3,136 x 4,224 píxeles y un peso de 4.04 MB, por tal motivo se requirió redimensionarlas a un tamaño de 768 x 1034

pixeles con la finalidad de reducir su peso para futuras aplicaciones, las imágenes resultantes después de esta modificación redujeron su peso a solo 150 KB.

Posteriormente, mediante el uso del software *Labelimg*, el cual es un programa destinado al etiquetado de imágenes por medio de un cuadro delimitador que indica el objeto que queremos detectar y asignar el nombre de su etiqueta, se realizó el correspondiente etiquetado de cada una de las 10,000 imágenes de la base de datos para asignarle su respectiva clase (Ilustración 52).



Ilustración 52 Etiquetado de la base de datos

4.4.4 Paso 4: Generación de la base de datos

Finalmente, en la Ilustración 53 se puede observar un ejemplo de cada una de las 10 clases de la base de datos generada, de esta forma, cada imagen junto con su etiqueta fue almacenada en su respectiva carpeta



Ilustración 53 Base de datos propia

4.5 Etapa 5: Protocolo de entrenamiento con transferencia de aprendizaje a la base de datos generada

La base de datos generada cuenta con una menor cantidad de imágenes, 1,000 imágenes por cada clase, en comparación con la base de datos “*State Farm Distracted Driver Detection*” que cuenta con 10,000 imágenes por clase y con la que se desarrollaron entrenamientos desde cero, por este motivo, dada la reducción en la

cantidad de datos es que se optó por utilizar la “transferencia de aprendizaje”, técnica que ayuda a mitigar la desventaja que se tiene al contar con un número menor de muestras para poder realizar entrenamientos desde cero.

El protocolo propuesto para los entrenamientos de los modelos preentrenados se muestra en la Ilustración 54.

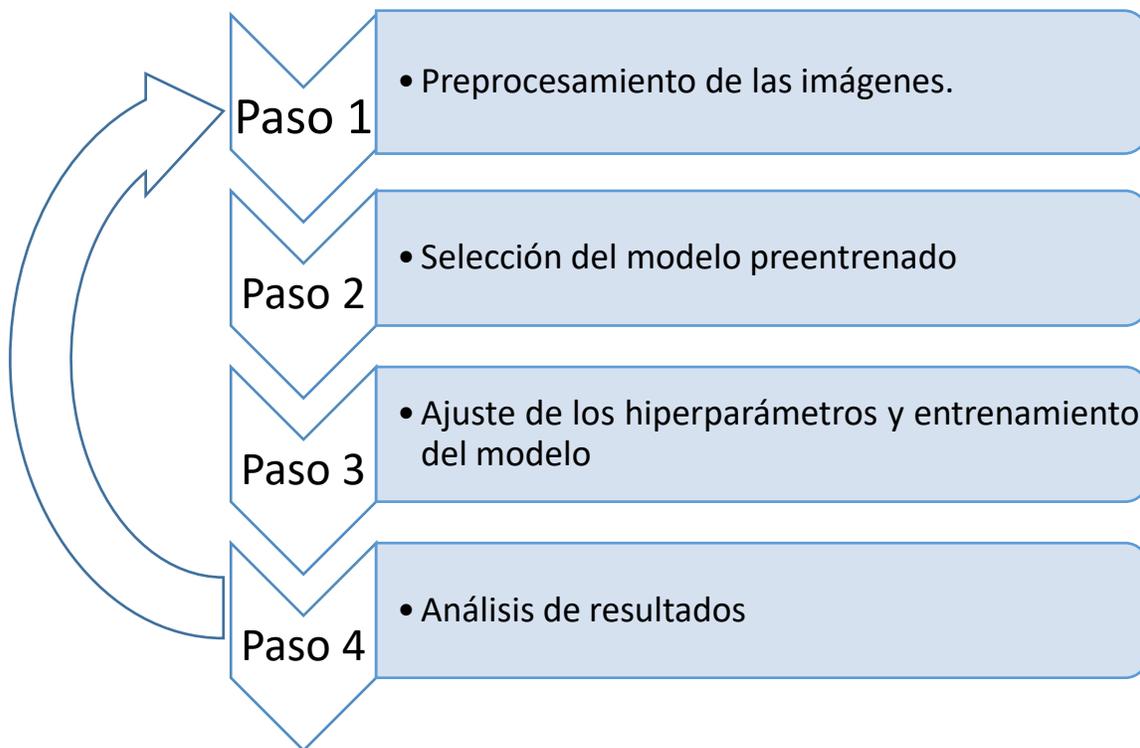


Ilustración 54 Protocolo de entrenamiento de modelos preentrenados

Salvo por la selección de los modelos preentrenados en el paso 2, el protocolo es casi similar al presentado para los entrenamientos de los modelos entrenados desde cero.

4.5.1 Paso 1: Preprocesamiento de las imágenes

De la misma manera que en los entrenamientos anteriores, se utilizó la técnica de las redes convolucionales para la clasificación multiclase y de igual forma se requirió realizar un nuevo redimensionamiento del conjunto de datos, retomando que la nueva base de datos cuenta con imágenes de unas dimensiones de 768 x 1024 fue necesario redimensionarlas a un tamaño de 384 x 384 (Ilustración 55). Este nuevo redimensionamiento es debido a que con estas dimensiones trabajan los modelos preentrenados, además de tener una menor pérdida de información al redimensionarlas a un tamaño menor como en el caso de los entrenamientos desde cero.

Se dividió el conjunto de datos en grupos de entrenamiento, con un 80% de los datos, y en validación, con un 20% de los datos, para realizar un mejor entrenamiento de la red y evitar los problemas comunes tal y como lo es la memorización.



Redimensionamiento de la imagen



768x1024



384x384

Ilustración 55 Redimensionamiento de la base de datos local

Es preciso señalar que para los modelos de entrenamiento de esta base de datos se tuvieron que trabajar con las imágenes en sus tres canales RGB debido a que es un requisito para poder trabajar con los modelos preentrenados seleccionados, los cuales reciben en su entrada datos con las dimensiones anteriormente mencionadas y en los 3 canales RGB, aunque esto conlleva un incremento en el costo computacional y en los tiempos de los entrenamientos es un requisito necesario para poder obtener la mayor cantidad de información posible de las imágenes.

4.5.2 Paso 2: Selección de modelos preentrenados

En esta ocasión se decidió trabajar con 3 modelos preentrenados desarrollados por Google dedicados a la detección y monitoreo de objetos, herramienta utilizada en la ejecución del proyecto, también se retomaron los entrenamientos con la red “MobileNet-v2” cuya principal función es la clasificación de imágenes.

La red preentrenada “Efficientnetb0” es una red neuronal convolucional, la cual fue entrenada con más de un millón de imágenes para la clasificación de 1000 categorías de objetos diferentes, debido a la numerosa cantidad de imágenes y de parámetros entrenables de la red es que esta misma ha aprendido características representativas para una amplia gama de imágenes, su arquitectura se muestra en la Ilustración 56 (MATLAB, 2023).

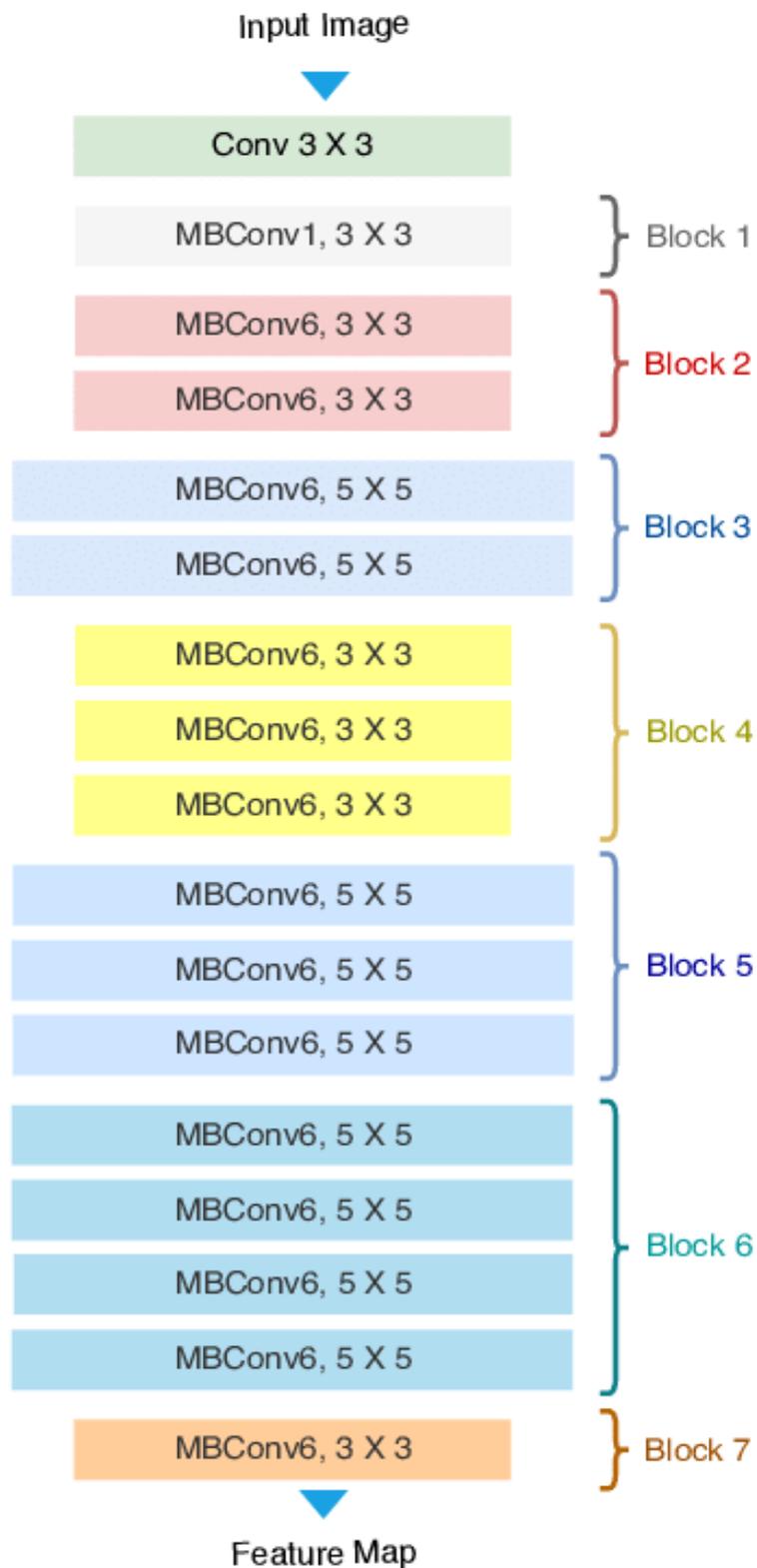


Ilustración 56 Arquitectura de la red "Efficientnetb0" obtenido de (Keras, 2023).

La red preentrenada “Efficientnetb1” es una red neuronal convolucional entrenada para la clasificación de 1000 clases, comparte características similares que la red anterior salvo por la cantidad de parámetros entrenables que conlleva su arquitectura más profunda vista en la Ilustración 57 (Keras, 2023).

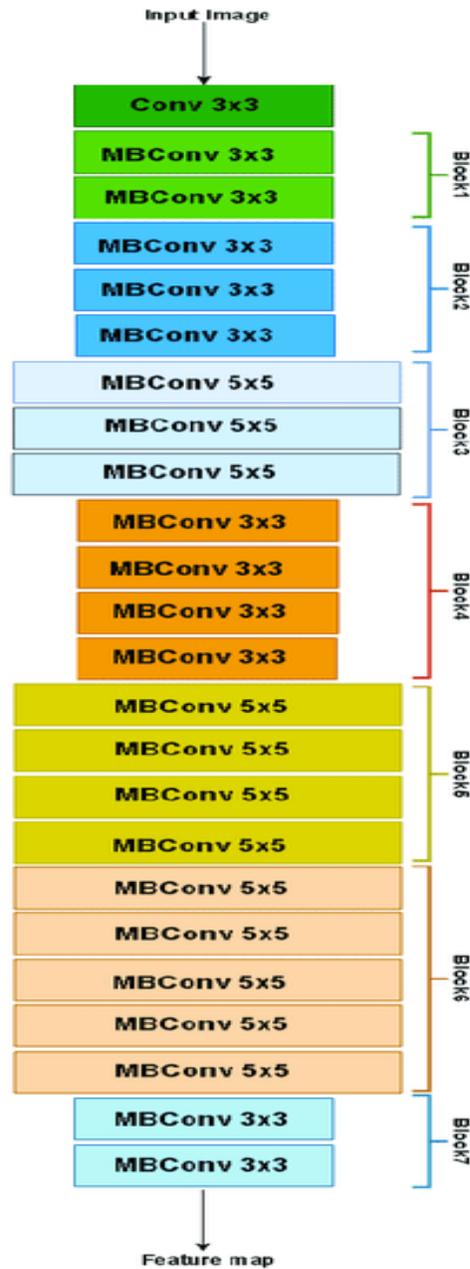


Ilustración 57 Arquitectura de la red “Efficientnetb1” obtenido de (Keras, 2023).

La red preentrenada “MobileNet-v2” ya fue examinada en la Ilustración 44 durante los entrenamientos con la base de datos “*State Farm Distracted Driver Detection*”.

4.5.3 Paso 3: Ajuste de hiperparámetros y entrenamiento aplicando transferencia de aprendizaje a la base de datos generada

De la misma forma en que se seleccionaron los hiperparámetros en los entrenamientos desde cero se siguió la propuesta sugerida por Simon (2009), en esta ocasión se trabajó primero con unos hiperparámetros un poco más grandes que los sugeridos y utilizados en entrenamientos anteriores con la finalidad de observar los resultados obtenidos en esta selección.

En la Tabla 23 se presentan los hiperparámetros seleccionados para entrenar los nuevos modelos preentrenados.

Tabla 19 Hiperparámetros para los primeros 4 modelos con la base de datos local

Hiperparámetros	
Tasa de aprendizaje	0.003
Tamaño del lote	200
Número de épocas	30

Dentro de los primeros cuatro entrenamientos utilizando la transferencia de aprendizaje con redes preentrenadas y utilizando los hiperparámetros de la Tabla 23 se obtuvieron los resultados observados en la Tabla 24.

Tabla 20 Resultados de los primeros 4 entrenamientos

	Efficientnetb0	Efficientnetb1	Efficientnetb2	MobileNet-v2
Exactitud	0.63	0.83	0.80	0.64
Exactitud_Val	0.65	0.87	0.83	0.65

Se puede observar que el modelo de red “Efficientnetb1” fue el modelo entrenado que obtuvo mejores resultados, el cual alcanzó un 83% de exactitud en el entrenamiento y un 87% en su validación, sin embargo, es un porcentaje por debajo del obtenido en los entrenamientos desde cero, por tal motivo se realizó una modificación en los hiperparámetros, en esta ocasión se optó por disminuir el valor de los mismos al tratar de acercarlos a los valores con los que se obtuvieron mejores resultados en las pruebas de los modelos entrenados desde cero, los nuevos hiperparámetros se muestran en la Tabla 25.

Tabla 21 Nuevos hiperparámetros para base de datos local

Hiperparámetros	
Tasa de aprendizaje	0.001
Tamaño del lote	150
Número de épocas	20

Los últimos cuatro experimentos realizados con la modificación de los hiperparámetros a las redes preentrenadas obtuvieron un mejor desempeño alcanzando los resultados presentados en la Tabla 26.

Tabla 22 Nuevos entrenamientos con nuevos hiperparámetros en la base local

	Efficientnetb0	Efficientnetb1	Efficientnetb2	MobileNet-v2
Exactitud	0.64	0.91	0.84	0.75
Exactitud_Val	0.69	0.95	0.88	0.82

Nuevamente la red “Efficientnetb1” utilizando la “transferencia de aprendizaje” obtuvo el mejor resultado dentro de los cuatro modelos de redes preentrenadas propuestos, alcanzando un resultado del 91% de exactitud en el entrenamiento y un 95% en la validación, siendo el mejor resultado utilizando la base de datos generada, lo cual es equiparable con el modelo anteriormente entrenado con la base de datos

“State Farm Distracted Driver Detection” y diseñando arquitecturas de modelos de redes propias, pero con una mayor cantidad de imágenes en la base de datos.

4.5.4 Paso 4: Evaluación del modelo con mejores resultados

La red convolucional preentrenada “Efficientnetb1” obtuvo una exactitud del 91% en la clasificación de la base de datos generada por lo que se procedió a examinar sus respectivas gráficas de exactitud y de pérdida durante el entrenamiento, como se explicó anteriormente, esto se realiza para garantizar que no haya ocurrido algún error durante los entrenamientos o que el modelo se encontrara memorizando en vez de aprender y de generalizar los datos de entrada. La gráfica que representa su curva de exactitud se puede observar en la Ilustración 59.

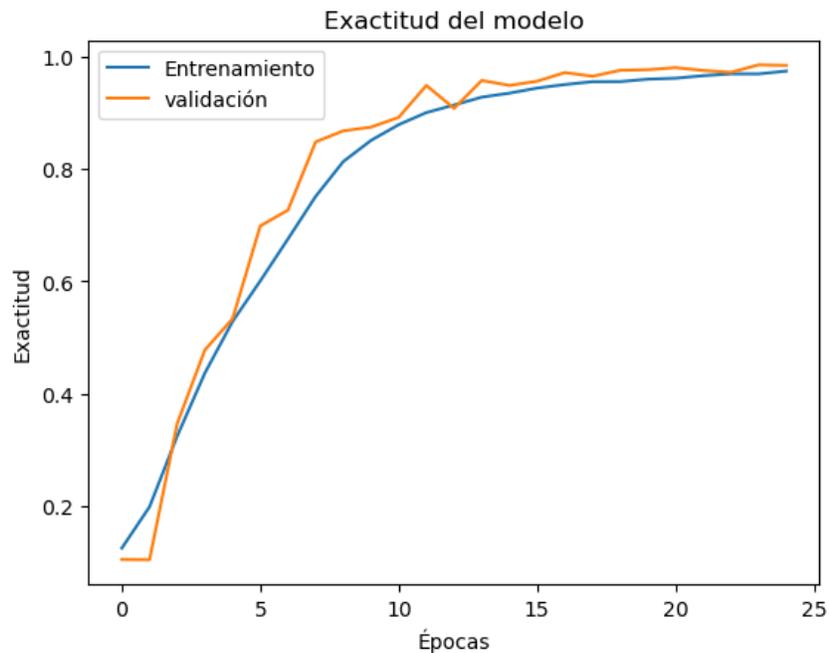


Ilustración 59 Exactitud del modelo con la base de datos generada

De la misma manera se examinó la gráfica de pérdida del modelo en la Ilustración 60, en donde se puede apreciar que las curvas de exactitud y de pérdida tanto en el entrenamiento como en la validación llevaron un correcto entrenamiento y aprendizaje por parte de la red.

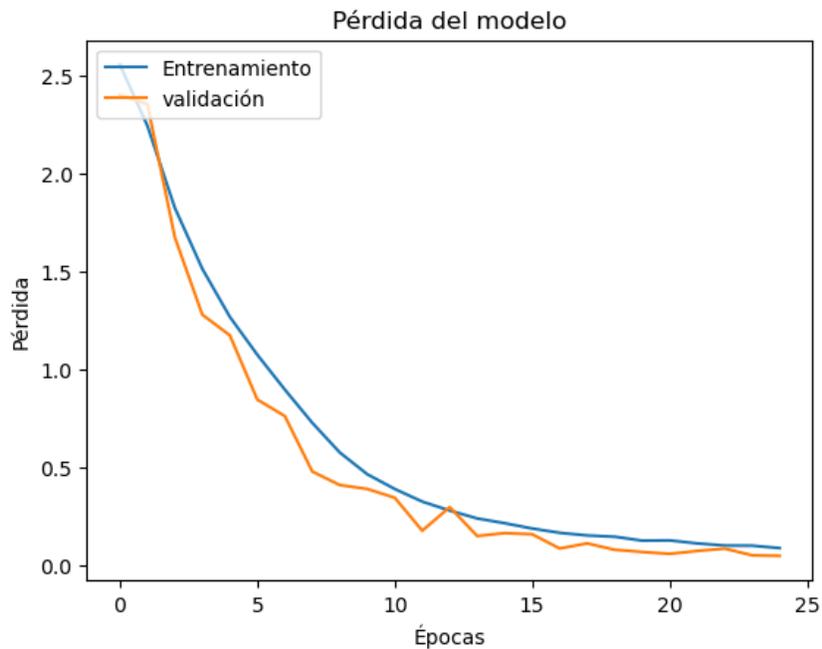


Ilustración 60 Pérdida del modelo con la base de datos generada

4.6 Etapa 6: Implementación móvil

Para la implementación del proyecto en dispositivos móviles se tomó como base los dispositivos con sistema operativo *Android*, ya que es el sistema operativo para celulares más estandarizado, además de pertenecer a *Google*, la empresa creadora de *TensorFlow*, una herramienta para los proyectos de aprendizaje automático que

servió de utilidad para poder convertir nuestro proyecto en uno compatible con estos dispositivos, lo cual otorga un soporte continuo para la plataforma y una serie de librerías que no están estandarizadas para todos los dispositivos.

Los pasos a seguir para realizar la conversión del modelo en formato *TensorFlow* a *TensorFlow Lite* se describen en la Ilustración 61.

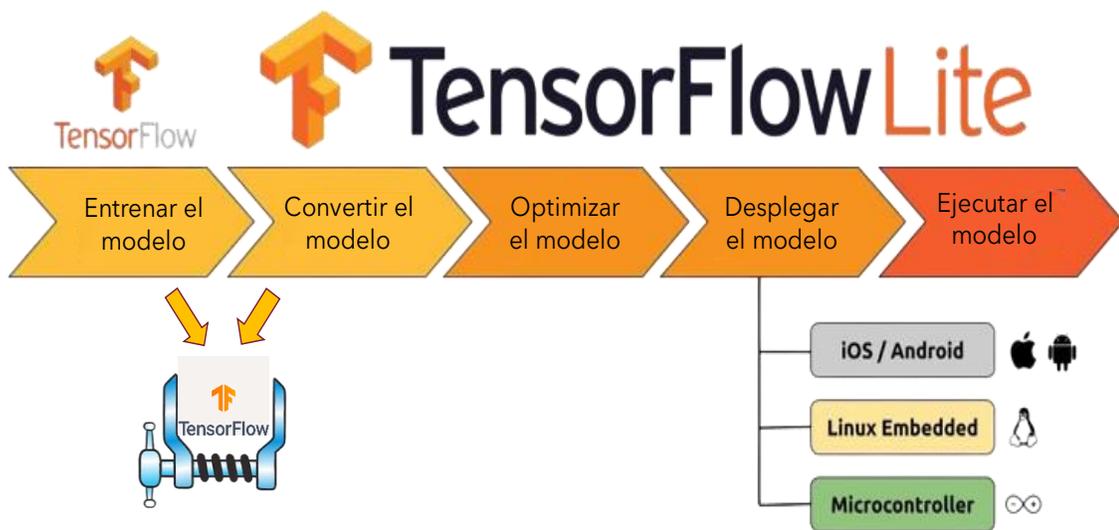


Ilustración 61 Conversión del modelo a TensorFlow Lite

Como lo indica la Ilustración anterior, antes de proceder al desarrollo de la aplicación es importante realizar algunos pasos extra en nuestro modelo de aprendizaje, se comienza cargando el modelo de entrenamiento con mejores resultados en un nuevo cuaderno de programación en *Python*, luego se procede a convertir el modelo en formato *TensorFlow Lite* (Ilustración 62), necesario para su ejecución en dispositivos móviles o en tarjetas de desarrollo.

Lo que se realiza con esta conversión es una disminución del tamaño del modelo original, lo que se realiza en esta etapa es eliminar cálculos redundantes y fusionar múltiples operaciones en un solo proceso (reducción de capas de entrenamiento) para aminorar la cantidad de parámetros entrenables por la red y disminuir su peso para que pueda ser instalado sin complicaciones por el tamaño que pueda abarcar en la memoria del dispositivo móvil y los recursos que pueda llegar a necesitar del mismo.

```
import tensorflow as tf
#store .h5 file in your .py folder

#load h5 module
model=tf.keras.models.load_model('/content/gdrive/MyDrive/ML_entrenamiento/Distracciones_90.h5')
tflite_converter = tf.lite.TFLiteConverter.from_keras_model(model)

#convert
tflite_model = tflite_converter.convert()
open("tf_lite_model.tflite", "wb").write(tflite_model)
```

Ilustración 62 Conversión del modelo a formato TF Lite

La desventaja generada durante esta conversión radica en que conlleva una pérdida en la exactitud alcanzada por el entrenamiento de la red preentrenada, de entre el 10 al 15 por ciento, motivo por el cual es necesario llevar a cabo una optimización del modelo, la cual consiste en “reentrenar” la red, pero esta vez en su versión en *TensorFlow Lite* (Ilustración 63).

```
In [11]: model.evaluate(val_data)
1/1 [=====] - 6s 6s/step

Out[11]: {'AP': 0.73879343,
          'AP50': 1.0,
          'AP75': 1.0,
          'APs': -1.0,
          'APm': -1.0,
          'APl': 0.73879343,
          'ARmax1': 0.7625,
          'ARmax10': 0.765,
          'ARmax100': 0.765,
          'ARs': -1.0,
          'ARm': -1.0,
          'ARl': 0.765,
          'AP_/Conduccion segura': 0.7708411,
          'AP_/Enviando SMS mano derecha': 0.7067457}
```

Ilustración 63 Evaluación de la aplicación

Se puede apreciar que los modelos exportados a un formato compatible con dispositivos móviles o tarjetas de desarrollo sufren una pérdida considerable en la exactitud de la red previamente entrenada, en este caso de un modelo que alcanzaba el 95% de exactitud en su entrenamiento terminó por reducirse a un 80%. Sin embargo, gracias a la optimización que se llevó a cabo con el reentrenamiento de la red, el modelo terminó obteniendo un 90% de exactitud.

Posteriormente es necesario instalar la librería “*tflite_support_nightly*”, el cual es un conjunto de herramientas que ayudan en la implementación del formato *TensorFlow Lite* en dispositivos móviles, esto es de utilidad para la importación de los metadatos, los cuales son archivos asociados al modelo como el archivo que contiene el nombre de las clases (las etiquetas) de nuestro programa.

Luego se une el modelo con los metadatos antes mencionados, tal y como se observa en la Ilustración 64.

```
from tflipe_support.metadata_writers import image_classifier
from tflipe_support.metadata_writers import writer_utils

ImageClassifierWriter = image_classifier.MetadataWriter

_MODEL_PATH = "/content/tflipe_model_another.tflipe"
_LABEL_FILE = "/content/gdrive/MyDrive/ML_entrenamiento/labelmap.txt"
_SAVE_TO_PATH = "/content/gdrive/MyDrive/ML_entrenamiento/metadistracciones.tflipe"

# Create the metadata writer.
writer = ImageClassifierWriter.create_for_inference(
    writer_utils.load_file(_MODEL_PATH), [_INPUT_NORM_MEAN], [_INPUT_NORM_STD], [_LABEL_FILE])

# Verify the metadata generated by metadata writer.
print(writer.get_metadata_json())

# Populate the metadata into the model.
writer_utils.save_file(writer.populate(), _SAVE_TO_PATH)
```

Ilustración 64 Unión del modelo con los metadatos

Para exportar nuestro modelo a una aplicación se utilizó el software *Android Studio*, por ser el entorno de desarrollo integrado (IDE por sus siglas en inglés) oficial para el sistema operativo *Android* de *Google*. Dentro de nuestro proyecto de la aplicación se agregó el modelo de la red TF Lite en la carpeta “*assets*” en donde se encontrará el modelo de clasificación junto con las etiquetas de las clases (Ilustración 65).

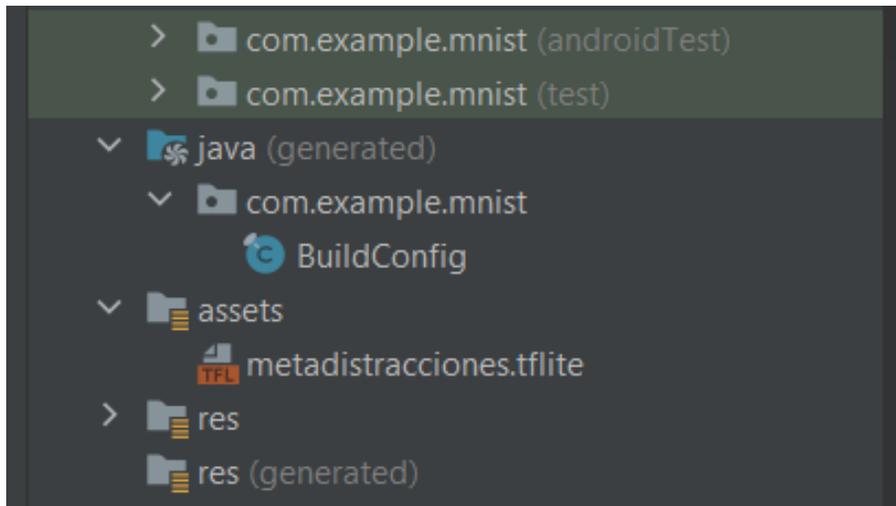


Ilustración 65 Carga del modelo TF Lite en la aplicación

Para el desarrollo de la respuesta del Sistema de Asistencia a la Conducción fue necesario grabar en audio la recomendación “Por favor no te distraigas” como una forma de alertar al conductor cada que el sistema detecte una de las nueve distracciones con las que fue entrenada.

- Enviando SMS con mano derecha
- Llamada con mano derecha
- Enviando SMS con mano izquierda
- Llamada con mano izquierda
- Manipulando la radio/GPS
- Ingeriendo bebidas
- Buscando algo en la parte de atrás
- Maquillando/Arreglando el cabello
- Hablando con pasajero

Este archivo de audio fue agregado a una carpeta creada dentro de la aplicación llamada “raw” (Ilustración 66).

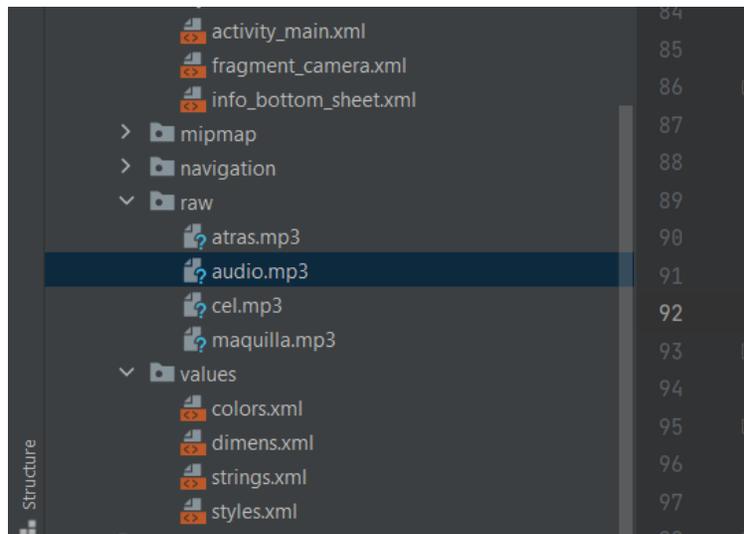


Ilustración 66 Recomendación del sistema

Por último, se conectó el dispositivo móvil a la computadora (en modo desarrollador) y se instaló la aplicación.

4.7 Etapa 7: Aplicación en funcionamiento

Al desplegar la aplicación en el dispositivo móvil (Ilustración 67) el sistema accede a la cámara del dispositivo y comienza a convertir las imágenes capturadas por la cámara en imágenes de entrada para el modelo, el cual examinará y clasificará los datos de entrada. Las salidas de la red se acompañan de un cuadro delimitador encargado de localizar al conductor y reconocer la acción que esté realizando, al tiempo que determina la etiqueta y la probabilidad de que la imagen coincida con esta misma.

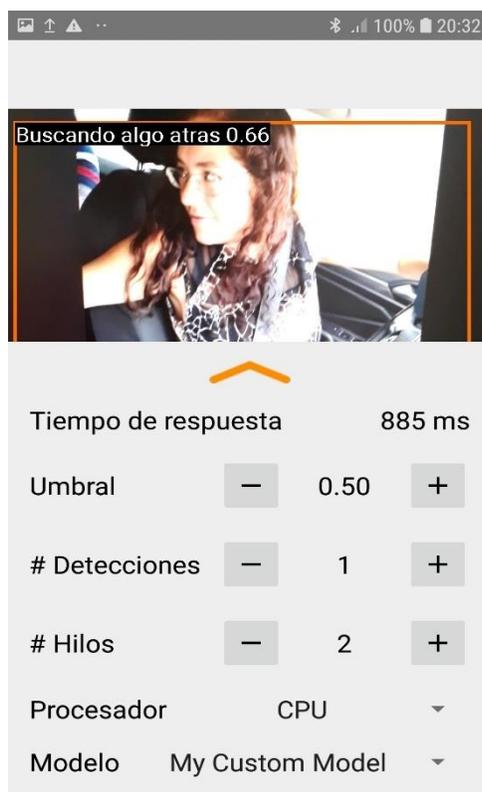


Ilustración 67 Aplicación en funcionamiento

El Sistema de Asistencia a la Conducción ofrece un tiempo de respuesta promedio de 885 milisegundos, dependiendo del modelo de dispositivo móvil, un tiempo considerablemente bueno para poder prevenir al conductor de algún indicio de distracción. Cabe destacar que el sistema es capaz de mejorar su rendimiento al modificar algunas de las opciones predeterminadas de la aplicación, tal y como poder acceder a la GPU en los celulares que cuenten con ella, poder utilizar más hilos de ejecución en el programa y también aumentar o disminuir el umbral de detección.

Capítulo V: Resultados y discusión

Al trabajar con problemas que involucren utilizar técnicas de aprendizaje profundo para la solución de problemas de clasificación y detección en imágenes, se deben de considerar aspectos importantes en el diseño de la solución, tales como la ambigüedad existente entre las clases (en problemas multiclase), la elección de la base de datos y la cantidad de imágenes que contienen estos conjuntos de datos. Cada uno de estos aspectos es de vital importancia para poder hallar un modelo óptimo que sea capaz de adaptarse a nuestras necesidades.

Las distracciones en los conductores de automóviles son un severo problema en nuestra sociedad y una de las principales causas de muerte a nivel mundial, por estos motivos se ha estado trabajando en la búsqueda de una solución mediante los Sistemas de Asistencia a la Conducción, que logren aminorar este alto índice de siniestros viales, destacando que esta tarea es de una alta complejidad para los sistemas inteligentes, incluso para nosotros mismos (los humanos) resulta un tema complejo de analizar.

Debido a la complejidad del problema y a la alta similitud entre las distracciones (las clases del modelo), se puede considerar que un modelo con un porcentaje de exactitud del 80% o superior es sinónimo de un modelo aceptable y esto se ve reflejado en el análisis que se desarrolló al examinar las gráficas de las curvas de aprendizaje de la exactitud y la pérdida de los mejores modelos, en las cuales se apreció que la

cantidad de errores en la clasificación fue minoritaria si se compara con la cantidad total de los datos, ambas gráficas.

De esta manera se hace hincapié en que se obtuvo un sistema equiparable en resultados con los presentados en la sección de antecedentes, cuyos sistemas reportan entre un 80 a 90 por ciento de exactitud, además el propósito innovador del proyecto es el poder implementar un Sistema de Asistencia a la Conducción implementando redes neuronales convolucionales en un dispositivo móvil, lo cual posibilita que cualquier conductor con un dispositivo móvil *Android* pueda instalar esta aplicación y ejecutar el asistente sin la necesidad de cambiar su vehículo por uno de gama alta con este tipo de sistemas, lo cual es una desventaja de los antecedentes comerciales, cuyos sistemas están adaptados a determinadas líneas de automóviles.

A modo de resumen, en las Tablas 27 y 28 se muestra una comparación entre los sistemas propuestos por las empresas comerciales distribuidoras de automóviles y el Sistema de Asistencia a la Conducción desarrollado en este proyecto.

En ambas tablas se logra apreciar que las empresas automotrices se han enfocado en robustecer sus sistemas utilizando técnicas complejas como el sensado del conductor y el uso de cámaras especializadas para enfocarse principalmente en la detección temprana de la somnolencia y en el desvío de la mirada. En desventaja, estos sistemas robustos son accesibles únicamente en ciertos automóviles de dichas empresas.

Tabla 23 Comparación con antecedentes comerciales 1

Proyecto	Distracciones analizadas	Tecnología empleada	Recursos necesarios	Evaluación de la solución
Attention assist (Mercedes-Benz, 2019)	Fatiga y somnolencia	Monitoreo del volante	Sensado del volante	Detecta indicios de fatiga en determinados automóviles
Smart eye (Smart eye, 2021)	Distracción y somnolencia	Inteligencia artificial	Cámara infrarroja especializada	Detecta indicios de fatiga y distracción en determinados automóviles
Sistema de Asistencia a la Conducción en dispositivo móvil	Detección de 9 distracciones	Inteligencia artificial y aprendizaje profundo	Dispositivo móvil	Detecta la conducción segura, 9 distracciones y se adapta a cualquier automóvil.

Tabla 24 Comparación con antecedentes comerciales 2

Proyecto	Distracciones analizadas	Tecnología empleada	Recursos necesarios	Evaluación de la solución
Driver Attention Monitoring (CARMANÍA, 2020)	Distracción y somnolencia	Visión artificial, seguimiento y monitoreo del conductor	Dos cámaras infrarrojas especializadas	Detecta indicios de fatiga y distracción en determinados automóviles
Tecnología ADAS Peugeot (Peugeot, 2020)	Distracción y somnolencia	Notificador de tiempo de conducción y monitoreo de líneas de carretera	Cámara externa	Detecta indicios de fatiga y distracción en determinados automóviles
Sistema de Asistencia a la Conducción en dispositivo móvil	Detección de 9 distracciones	Inteligencia artificial y aprendizaje profundo	Dispositivo móvil	Detecta la conducción segura, 9 distracciones y se adapta a cualquier automóvil.

Las Tablas 29 y 30 realizan una comparación entre el ADAS desarrollado y las propuestas de investigación desarrolladas por la comunidad científica.

Tabla 25 Comparación con antecedentes de investigación 1

Proyecto	Distracciones analizadas	Tecnología empleada	Recursos necesarios	Evaluación de la solución
Asistente de conducción de detección de somnolencia (Flores, 2020)	Fatiga y somnolencia	Monitoreo del rostro y apertura de los ojos	Cámara	Detecta indicios de fatiga adaptable en automóviles
Monitorización del conductor en Android (Salcedo, 2015)	Distracción por desvío de mirada	Visión artificial	Cámara del dispositivo móvil	Detecta indicios de distracción adaptable en automóviles
Sistema de Asistencia a la Conducción en dispositivo móvil	Detección de 9 distracciones	Inteligencia artificial y aprendizaje profundo	Dispositivo móvil	Detecta la conducción segura, 9 distracciones adaptable a cualquier automóvil.

Tabla 26 Comparación con antecedentes de investigación 2

Proyecto	Distracciones analizadas	Tecnología empleada	Recursos necesarios	Evaluación de la solución
Detección de emociones (Romero, 2020)	Detección de emociones	Aprendizaje profundo	Cámara	Detecta emociones, pero no es preciso.
Sistema de detección de fatiga en conductores (Paterna, 2020)	Detección de fatiga y somnolencia	Visión artificial, monitoreo de apertura de ojos y boca	Cámara	Detecta indicios de somnolencia adaptable en automóviles
Sistema de Asistencia a la Conducción en dispositivo móvil	Detección de 9 distracciones	Inteligencia artificial y aprendizaje profundo	Dispositivo móvil	Detecta la conducción segura, 9 distracciones adaptable a cualquier automóvil.

Las soluciones planteadas por la comunidad científica son capaces de detectar una o dos distracciones en los conductores, pero requieren el uso de recursos extra

tales como cámaras y microcontroladores encargados de recibir los datos de entrada de la cámara y llevar a cabo los trabajos de detección, algunos de ellos continúan en etapas de pruebas, por tal motivo todavía no cuentan con suficiente precisión.

Capítulo VI: Conclusiones

En el presente proyecto de investigación se desarrolló un Sistema Avanzado de Asistencia a la Conducción cuya finalidad es poder contribuir en la búsqueda de sistemas capaces de detectar y alertar a los conductores la presencia de la conducción distraída, misma que es una de las principales causas de los accidentes viales y que se puede presentar de diversas formas (diversas distracciones), debido a la importancia del tema de la seguridad vial es que se han buscado formas de detectar distracciones que propicien un mal manejo de los vehículos, por ello este proyecto tiene como finalidad poder asistir a los conductores de automóviles mediante la detección de 9 distracciones y poder alertarlo de una conducción distraída a través de una aplicación para dispositivos móviles, de esta manera se concluye que la hipótesis ha sido comprobada y los objetivos del proyecto fueron cumplidos, destacando así el uso de la inteligencia artificial, el aprendizaje profundo y las redes neuronales convolucionales y su empleo en el desarrollo de proyectos que cubran problemas sociales como el de la conducción segura y la seguridad vial.

6.1 Trabajo a futuro

El trabajo de investigación presentado abre las puertas para el desarrollo de investigaciones y trabajos futuros experimentales, bajo las líneas de investigación de los Sistemas Avanzados de Asistencia a la Conducción, la detección de la conducción distraída y las técnicas avanzadas de aprendizaje profundo.

Algunos de los trabajos que se tienen contemplados para trabajar en un futuro se enlistan a continuación.

- Realizar un análisis de las distracciones temporales y no temporales en los conductores de automóviles.
- Probar la incorporación de una base de datos multimodal (audio y video) de conductores realizando distracciones temporales y no temporales.
- Realizar entrenamientos con redes recurrentes para la detección y clasificación de distracciones temporales
- Realizar un cambio de dominio en el aprendizaje de una red transformer para solventar la escases de datos que se tiene de conductores realizando actividades riesgosas (somnolencia, estado de ebriedad, tabaquismo, etc.)
- Analizar y experimentar la modificación de redes preentrenadas (transferencia de aprendizaje) para adaptarlas a un problema en específico y para incrementar su exactitud en la clasificación de imágenes.
- Diseño de mensajes vocales de advertencia para la aplicación ADAS con técnicas de experiencia de usuario (UX).

REFERENCIAS

- Aggarwal, C. C. (2018). *Neural Networks and Deep Learning: A Textbook*. Springer Nature.
- Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.
- Álvarez, C. G. (Septiembre de 2015). *Adaptación de las Metodologías Tradicionales Cascada y Espiral*. Obtenido de Tesis: <http://repositorio.utm.mx/bitstream/123456789/72/1/2015-MMI-CGPA.pdf>
- Arbib, M. A. (2019). *Handbook of brain theory and neural networks*. The MIT Press.
- Azuela, J. H. (2013). *Visión Artificial: Rasgos Descriptores para el Reconocimiento de Objetos*. España: RA-MA.
- Behnke, S. (2003). *Hierarchical Neural Networks for Image Interpretation*. Springer.
- Berzal, F. (2019). *Redes neuronales & deep learning: Volumen ii*. Independently published.
- Burch, C. (2001). *A survey of machine learning*. A survey for the Pennsylvania.
- Calero, M. J., Moreno, J. M., & Hueso, A. d. (2010). Sistema Avanzado de Asistencia a la Conducción para la Detección de la Somnolencia. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 216-228.
- CARMANÍA. (Junio de 2020). *Innovador sistema DS Driver Attention Monitoring*. Obtenido de CARMANÍA: <https://carmania.mx/2020/06/08/innovador-sistema-ds-driver-attention-monitoring/>
- Carro Pérez, E. H. (2019). *Conductas de riesgo al conducir un automóvil en zonas urbanas del sur de Tamaulipas y la Ciudad de México*. Obtenido de CienciaUAT: <https://doi.org/10.29059/cienciauat.v13i2.988>
- Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.
- Chris, S., & Toby, B. (2011). *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*. Wiley-Blackwell.
- Clínica Rahhal oftalmología. (2023). *Vista cansada*. Obtenido de Clínica Rahhal oftalmología: <https://www.rahhal.com/edad-comienzan-sintomas-vista-cansada/>
- Comisión Nacional de Seguridad. (Septiembre de 2015). *Accidentes y sus Factores*. Obtenido de Comisión Nacional de Seguridad: http://www.cns.gob.mx/portalWebApp/appmanager/portal/desk?_nfpb=true&_pageLabel=portals_portal_page_m2p1p2&content_id=830068&folderNode=830052&folderNode1=81027

- DeepChecks. (2022). *Learning Rate in Machine Learning*. Obtenido de DeepChecks: <https://deepchecks.com/glossary/learning-rate-in-machine-learning/>
- Dirección General de Tráfico. (Septiembre de 2020). *Los peligros de hablar por el celular*. Obtenido de Dirección General de Tráfico (DGT): <https://www.autobild.es/noticias/dgt-resume-rapido-peligros-hablar-130-minutos-movil-719381>
- Dirección General de Tráfico de España. (2023). *Conciencia Vial*. Obtenido de Dirección General de Tráfico de España: www.dgt.es
- Ekman, M. (2021). *Learning Deep Learning: Theory and Practice of Neural Networks, Computer Vision, Natural Language Processing, and Transformers Using Tensorflow*. Addison-Wesley Professional.
- Fabio, S., & Manuela, C. (2012). *Machine Vision - Applications and Systems*. INTECH.
- Flórez, L. Á. (Octubre de 2020). *Reconocimiento de emociones y comportamientos en conductores vía Deep Learning*. Obtenido de Trabajo Fin de Grado: https://e-archivo.uc3m.es/bitstream/handle/10016/32287/TFG_Laura_Alvarez_Florez.pdf?sequence=1&isAllowed=y
- FUNDACIÓN Carlos Slim. (Agosto de 2016). *Distracciones al conducir – Seguridad Vial*. Obtenido de FUNDACIÓN Carlos Slim: <https://fundacioncarlosslim.org/distracciones-al-conducir-seguridad-vial/>
- García, L. (Marzo de 2021). *El papel de la Inteligencia Artificial en la conducción autónoma*. Obtenido de El Economista: <https://revistas.eleconomista.es/digital/2021/marzo/el-papel-de-la-inteligencia-artificial-en-la-conduccion-autonoma-YG6822515>
- Gobierno del estado de Jalisco. (2022). *Gobierno del estado de Jalisco*. Obtenido de Recomendaciones para la prevención de accidentes viales: <https://www.jalisco.gob.mx/es/gobierno/comunicados/recomendaciones-para-la-prevencion-de-accidentes-viales-comparte-el-camino>
- Gomilla, J. (2019). *Deep learning de a a z: redes neuronales en python desde cero*. España: Udemy.
- González, F. A. (2015). *Modelos de aprendizaje computacional en reumatología*. Elsevier.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. The MIT Press.
- Goro, O., & Ashish, D. (2007). *Sistemas de Visión: Aplicaciones*. INTECH.
- Graupe, D. (2019). *Principles of artificial neural networks*. World Scientific.
- Harrington, P. (2012). *Machine learning in action*. Manning Publications Co.
- Haykin, S. (2009). *Neural Networks and Learning Machines*. New York: Prentice Hall/Pear-
- HUAWEI. (2023). *MACHINE LEARNING*. Obtenido de HUAWEI: <https://forum.huawei.com/enterprise/es/%C2%BFqu%C3%A9-es-epoch-en-machine-learning/thread/667232453749784577-667212895009779712>

- INEGI. (2020). *Accidentes de tránsito terrestre en zonas urbanas y suburbanas*. Obtenido de INEGI: https://www.inegi.org.mx/sistemas/olap/consulta/general_ver4/MDXQueryDatos.asp?#Regr eso&c=13159
- Instituto Mexicano del Transporte. (2022). *Instituto Mexicano del Transporte*. Obtenido de *Análisis de los límites de velocidad en carreteras*: <https://imt.mx/resumen-boletines.html?IdArticulo=116&IdBoletin=38>
- Instituto Nacional de Salud Pública. (Agosto de 2020). *México, séptimo lugar mundial en siniestros viales*. Obtenido de Instituto Nacional de Salud Pública: <https://www.insp.mx/avisos/4761-seguridad-vial-accidentes-transito.html>
- Isoba, M. C. (2021). *MANUAL PARA LA CONDUCCIÓN SEGURA*. Obtenido de Luchemos por la vida: <https://www.luchemos.org.ar/files/manualparaconduccionsegura.pdf>
- Jaramillo Romero, S. E. (2020). *Detección de emociones y reconocimiento facial utilizando aprendizaje profundo*. Obtenido de EXAMEN COMPLEXIVO: <http://repositorio.utmachala.edu.ec/handle/48000/15453>
- Kaggle. (2022). *Kaggle*. Obtenido de State Farm Distracted Driver Detection: <https://www.kaggle.com/competitions/state-farm-distracted-driver-detection/data>
- Keras. (2023). *EfficientNet*. Obtenido de Keras Applications: <https://keras.io/api/applications/efficientnet/#efficientnetb1-function>
- Konar, A. (2000). *Artificial intelligence and soft computing: behavioral and cognitive modeling of the human brain*. CRC Press.
- Lindblad, T., & Kinser, J. M. (2005). *Image Processing Using Pulse-Coupled Neural Networks*. Springer.
- Lipovac, K., Đerić, M., Tešić, M., Andrić, Z., & Marić, B. (2017). Mobile phone use while driving-literary review. *Transportation Research Part F: Traffic Psychology and Behaviour*, 47, 132–142. DOI: 10.1016/j.trf.2017.04.015.
- López, R. F., & Fernández, J. M. (2008). *Las Redes Neuronales Artificiales*. Netbiblo, S.L.
- MATLAB. (2023). *MathWorks*. Obtenido de MathWorks: <https://la.mathworks.com/discovery/transfer-learning.html>
- MATLAB. (2023). *Red neuronal convolucional EfficientNet-b0*. Obtenido de MathWorks: <https://la.mathworks.com/help/deeplearning/ref/efficientnetb0.html>
- MATLAB. (2023). *Red neuronal convolucional VGG-16*. Obtenido de MathWorks: <https://la.mathworks.com/help/deeplearning/ref/vgg16.html>
- Max, B., & Devedzic, V. (2004). *Artificial intelligence applications and innovations*. Boston, Mass: Kluwer Academic Publishers.
- Mercedes-Benz . (2019). *ATTENTION ASSIST*. Obtenido de Mercedes-Benz : <https://www.la.mercedes-benz.com/es/passengercars/mercedes-benz-cars/models/v-class/v-class->

mpv/explore.pi.html/mercedes-benz-cars/models/v-class/v-class-mpv/explore/intelligent-technologies/attention-assist

Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). *Machine Learning, Neural and Statistical Classification*. Prentice Hall.

Murguía, C. (2007). *Procesamiento digital de imágenes*. México: Editorial Trillas.

Núñez, R. V. (Junio de 2021). ANÁLISIS VISUAL DE CONDUCTOR IMPLEMENTANDO UNA RED NEURONAL PARA RECONOCER EMOCIONES Y DISTRACCIONES EN LA CONDUCCIÓN. *Comunicación personal*. Universidad Tecnológica de la Mixteca.

Or, J. (2011). *Affective Computing, Focus on Emotion Expression, Synthesis and Recognition*. InTech Education.

Organización Mundial de la Salud. (2019). *Informe sobre la situación mundial de la seguridad vial: es hora de pasar a la acción*. Obtenido de Organización Mundial de la Salud: https://www.who.int/violence_injury_prevention/road_safety_status/report/web_version_es.pdf?ua=1

Paterna, P. J. (2020). *Desarrollo de un sistema*. Obtenido de TRABAJO FIN DE ESTUDIOS: <https://repositorio.upct.es/xmlui/bitstream/handle/10317/7793/tfg-gar-des.pdf?sequence=1&isAllowed=y>

Peugeot. (2020). *TECNOLOGÍA PEUGEOT "ADAS"*. Obtenido de Peugeot: <https://media-ct-ndp.peugeot.com/file/04/2/presentaci-n-adas-2019-convertido.571042.pdf>

Planing, P. (2014). *Innovation Acceptance: The Case of Advanced Driver-Assistance Systems*. Springer.

Protección Civil. (2021). *Protección civil*. Obtenido de Y tú...¿Tienes distracciones al conducir?: <https://proteccioncivil.capitaldezacatecas.gob.mx/y-tutienes-distracciones-al-conducir/>

Rabunal, J. R., & Dorrado, J. (2005). *Artificial Neural Networks in Real-life Applications*. Idea Group Publishing.

Renault Eurofranceses de México. (Noviembre de 2021). *Principales causas de Accidentes*. Obtenido de RENAULT: <https://www.renaulteuro.com/blog/principales-causas-de-accidentes>

Roberto, R. M., & Juan, H. S. (2012). *Procesamiento y Análisis Digital de Imágenes*. Alfaomega.

Salcedo, Á. L. (2015). MONITORIZACIÓN DEL CONDUCTOR BASADO EN DISPOSITIVO ANDROID. *TRABAJO FIN DE GRADO*. Madrid.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510-4520, <https://doi.org/10.48550/arXiv.1801.04381>.

Secretaría de Salud/STCONAPRA. (2019). *Informe sobre la situación de la seguridad vial*. Obtenido de Secretaría de Salud/STCONAPRA: https://drive.google.com/file/d/1Y3jBmQqFBDuMok5rTGgO_87S4nVMIdRQ/view

- SEMOVI. (2022). *Reglamento de tránsito*. Obtenido de Reglamento de tránsito en carreteras federales: <https://mexico.justia.com/federales/reglamentos/reglamento-de-transito-en-carreteras-federales/titulo-iv/capitulo-ii/>
- Simon, S. H. (2009). *Neural networks and learning machines*. New York: Prentice Hall.
- Smart Eye . (Agosto de 2021). *Smart Eye y OmniVision desarrollan tecnología de seguridad para interiores de automóviles*. Obtenido de Ventas de seguridad: <https://www.ventasdeseguridad.com/2021081012821/noticias/empresas/smart-eye-y-omnivision-desarrollan-tecnologia-de-seguridad-para-interiores-de-automoviles.html>
- Toyota. (2018). *Volante con sensores para el ritmo cardíaco*. Obtenido de Toyota: <https://motorfull.com/2011/07/toyota-presenta-un-volante-con-sensores-para-el-ritmo-cardiaco>
- Vendrell, D. (Enero de 2021). *LeasePlan*. Obtenido de 5 distracciones al volante: medidas de prevención: <https://www.leaseplango.es/blog/seguridad-vial/5-distracciones-frecuentes-al-volante-aprende-a-evitarlas/>
- Volkswagen. (2020). *¿Por qué debes evitar manejar cansado en tu automóvil?* Obtenido de Volkswagen: <https://www.vw.com.mx/es/experiencia/tips/riesgos-de-manejar-cansado.html>
- Yildirim, G. (2010). *Affective computing and interaction: psychological, cognitive and neuroscientific perspectives*. IGI Global snippet.

Apéndice A

Publicaciones

En esta sección se presenta la portada del artículo aceptado y próximo a presentar en el XXV Congreso Mexicano de Robótica (COMRob 2023)

El artículo lleva por nombre “Neural networks for the detection of distractions in drivers”, se presentará en Noviembre del 2023 en Xalapa, Veracruz, México. El trabajo presentado en el artículo consiste en mostrar los resultados experimentales obtenidos tras realizar entrenamientos utilizando redes neuronales convolucionales mediante dos experimentos para la detección de la conducción distraída en un ADAS. El primer experimento utilizó arquitecturas de modelos entrenados desde cero con la base de datos descargada “*State Farm Distracted Driver Detection*”, el segundo experimento consistió en generar una base de datos propia para ser entrenada por modelos preentrenados, haciendo uso de la técnica “transferencia de aprendizaje”, obteniendo el mejor modelo en exactitud de la clasificación de las clases con el segundo experimento, mismo que por medio de la herramienta *TensorFlow Lite* fue implementado en una aplicación móvil para su ejecución en tiempo real.

Neural networks for the detection of distractions in drivers

1st Eduardo Aragón López
 División de estudios de Posgrado
 Universidad Tecnológica de la Mixteca
 Oaxaca, México
 0009-0009-2787-9282

2nd José Anibal Arias-Aguilar
 División de estudios de Posgrado
 Universidad Tecnológica de la Mixteca
 Oaxaca, México
 0000-0002-8838-877X

Abstract— Due to high rates of self-inflicted accidents, driving a car is a demanding activity that requires the driver to prepare both physically and mentally. In this sense, driving assistance systems play an important role in preventing distractions and road accidents. In this work, we detect and classify driver distractions using convolutional neural networks trained first with the “State Farm Distracted Driver Detection” and then with a custom database. Furthermore, with the use of the TensorFlow Lite library, the trained model is deployed on a mobile device for real-time use. The best trained model reaches an accuracy rate of 91% in training and 95% in validation, thus resulting in a highly discriminating model for each of the classes (distractions) to be classified. In other words, 10 different classes have been identified: one of them is “safe driving” and the other 9 correspond to those most responsible for accidents. With the system’s implementation on the mobile device, its performance is slightly reduced to 90% however it correctly detects distractions regardless of the class to which they belong. The initial results demonstrated the possibility of adapting an ADAS in a mobile device to detect distractions and generate real-time audible voice alerts.

Keywords—Neural networks, road accidents, driver, machine learning, mobile device, ADAS, object detection.

I. INTRODUCTION

According to [1], car accidents are the eighth-place cause of death globally, and Mexico ranks number seven in this list [2]. Concerning the countries of Latin America, our country occupies second place with the highest mortality rate due to traffic accidents [3]. It is important to note that 80% of road accidents are due to driver inattention because of distractions on the road [4].

According to [5], carrying out a second activity while driving adds extra demands on the driver’s attention, which causes him to divert his attention from critical activities for safe driving, thus possibly triggering a serious accident.

Therefore, distracted driving has become a principal theme of investigation for road safety. Research has been promoted in search of solutions that help mitigate this problem since it is broad, complex, and has not been sufficiently studied [6].

However, the industry has focused solely on developing solutions for higher-priced, more recent generation cars, known as high-end cars, to which very few people have access. Additionally, as [7] describes most of the proposed solutions are

not very efficient. Some of these are invasive and too complex to replicate. They can divert the driver’s attention to cause an accident; thus, causing more problems than they were looking to solve.

The company [8], for example, uses a fatigue detection system based on constant monitoring of the movement of the steering wheel. If the system detects an anomaly, the driver is recommended to take a break.

Similarly, [9] incorporates a lane keeping system in conjunction with a fatigue sensor. This complex system includes an exterior camera to monitor the delimiting lines of the road lane and a sensor in charge of monitoring the movement patterns of the steering wheel to indicate to the driver, by means of a light signal, about a possible mishap.

On the other hand, the technological solutions created so far by specialists are based on the implementation of artificial vision systems. For example, [10] implemented image descriptors and machine learning techniques to detect the presence (or absence) of the driver, to determine whether their eyes were open, or if they began to suffer early effects of drowsiness.

Reference [11] presented a fatigue and sleepiness detection system using artificial intelligence. In this case, a camera monitors the driver in real time to detect if they display a prolonged change in gaze deviation, or if their eyes close, with the objective to activate an audible alarm if so.

In [12], another alternative was introduced for the identification of fatigue, drowsiness and/or distraction in real time by means of optical sensors installed in the vehicle cabin. These sensors monitor head movements, closed eyes, and other unsafe behaviors of the driver, and launch an alarm when danger is detected.

Most of these investigations are focused on detecting a single distraction. Alternative (more complex) proposals require a higher cost in their production due to high-resolution cameras and powerful microcontrollers that carry out the detection. For this reason, an alternative system is proposed that applies deep learning techniques- implemented through an application for a mobile device- that detects and prevents some of the most common distractions while driving.

The rest of the article is structured as follows: section II presents the project participants, the justification of the

Apéndice B

Convocatorias

En esta sección se presenta la convocatoria a la cual fue sometido y aceptado el presente proyecto de tesis.

El proyecto de investigación “” fue postulado y aceptado en la convocatoria de “Proyectos de investigación en Inteligencia Artificial en el Espacio de Innovación UNAM-HUAWEI”, la cual fue una convocatoria resultado de la alianza entre la Universidad Nacional Autónoma de México (UNAM), Huawei Technologies, la Secretaría de Relaciones Exteriores (SRE), la Secretaría de Economía (SE), el Consejo Nacional para el Fomento a la Inversión, el Empleo y Crecimiento Económico (COFINECE), la Cooperación Alemana al Desarrollo Sustentable en México (GIZ) y la Red de Soluciones de Desarrollo Sostenible (SDSN) de la ONU, es una vinculación multisectorial, con el objetivo de consolidar una propuesta orientada a fortalecer el desarrollo, la investigación y la implementación de nuevas tecnologías, con énfasis en la inteligencia artificial (IA), para el desarrollo de propuestas de proyectos específicos para la solución de problemas sociales en México.

Proyectos de investigación en Inteligencia Artificial en el Espacio de Innovación UNAM – HUAWEI

Proyectos aprobados



RELACIONES
EXTERIORES



giz
Deutsche Gesellschaft
für Internationale
Zusammenarbeit (GIZ) GmbH



UNAM
La Universidad
de la Nación

La Universidad Nacional Autónoma de México
a través de la Coordinación de Vinculación y Transferencia Tecnológica

otorga la presente

Constancia

a

Eduardo Aragón López

por su participación en el "Customer Discovery para la Transferencia Tecnológica"
Impartido del 22 de mayo al 26 de junio del 2023 con duración de 18 horas.

"POR MI RAZA HABLARÁ EL ESPÍRITU"

Ciudad Universitaria, Cid. Ma., 28 de junio del 2023.

Mtro. Eduardo Uruza Fernández
Director de Emprendimiento Universitario

Innova UNAM
Emprendimiento

CVTI
Coordinación de
Vinculación y
Transferencia
Tecnológica

vinculación
UNAM