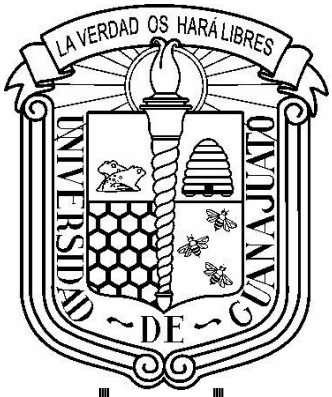

UNIVERSIDAD DE GUANAJUATO

CAMPUS IRAPUATO-SALAMANCA
DIVISIÓN DE INGENIERÍAS

Detección de objetos para navegación de vehículos autónomos basada en visión

TESIS

QUE PARA OBTENER EL TÍTULO EN:

INGENIERÍA EN MECATRÓNICA

PRESENTA:

Jonathan Duarte Jasso

DIRECTORES:

Dra. Dora Luz Almanza Ojeda

Dr. José Luis Contreras Hernández

SALAMANCA, GTO. Junio, 2022

Agradecimientos Personales.

A mis asesores de proyecto de tesis, la Dra. Dora Luz Almanza Ojeda y al Dr. José Luis Contreras Hernández, por su dedicación y la paciencia durante este proyecto.

A mi abuelita María Antonia Jasso Morales por sus enseñanzas que me ha brindado a lo largo de mi vida.

A mi madre Olga Lilia Duarte Jasso quien ha sido madre y padre a la vez, contribuyendo a mi desarrollo como persona y me ha impulsado a cumplir mis metas.

A mi tío el Lic. Alfonso Vallejo Esquivel, quien es una de las personas que más admiro en mi vida, el dejo este mundo estando orgulloso de mi ingreso a la Universidad de Guanajuato y hoy puedo decirle, se cumplió el objetivo.

A Eduardo Orozco Guzmán, quien ha sido un amigo y hermano, agradeciendo por compartir conmigo grandes experiencias.

A la Mtra. Ma. Susana Pérez Camacho, por las oportunidades brindadas durante su tutela, fortaleciendo a la comunidad estudiantil.

A la Mtra. Isabel Cano López por permitirme ser parte de los proyectos

de innovación para beneficio de los estudiantes.

A mi familia por siempre apoyarme y ser un gran pilar en mi formación

A mis amigos porque cada vivencia ha sido importante para afrontar lo que sigue en la vida.

A mis compañeros de trabajo y proyectos de los cuales he aprendido cosas nuevas.

A mi computadora HP por soportar algunos procesos de los entrenamientos de las RNC.

Índice general

Índice de figuras

Índice de Tablas

Resumen

El presente proyecto de tesis está basado en el área de la conducción de vehículos autónomos, que hoy en día es un tema de gran auge en la automatización de los medios de transporte y seguridad vial. Este tema tiene muchas sub-ramas, la que se tratará en este proyecto será en la detección de objetos, mediante una red neuronal convolucional entrenada para 11 clases, pertenecientes a diversos objetos que se encuentran en los ambientes urbanos. En la adquisición de datos se obtuvieron de ambientes reales, en los que se utilizó una máscara basada en regiones para limpiar la imagen de objetos que no pertenecieran a las 11 categorías, esto se hizo con la finalidad de tener un conjunto de datos preciso respecto al objeto de interés. Se tomaron en cuenta 6 diferentes redes neuronales convolucionales para entrenarlas con el conjunto de datos, el entrenamiento se tomó un 80 % del conjunto de datos y para la validación el 20 % restante. En el rendimiento de las redes neuronales convolucionales se mencionarán las tres redes con mejores resultados, en primera posición esta AlexNet, esta red fue capaz de obtener la distinción de las 11 clases en diferentes imágenes con un bajo porcentaje de confusión. La VGG19, la cual al igual que la red anterior, logró detectar las 11 categorías, pero con la diferencia de un porcentaje mayor de unidades para confundir la predicción final. Por último, la tercera red con mejor rendimiento fue la VGG16, la cual logró predecir 7 clases de las 11 que se tienen como predeterminadas con un 80 % de predicción por cada clase.

Capítulo 1

Introducción

La conducción autónoma ha evolucionado en los últimos años, durante el desarrollo de un sistema de dispositivos de muestreo, simultáneamente se ha desarrollado una mejora en la tecnología, logrando compaginar la innovación en la industria automotriz para conducción autónoma. Dentro de esta parte se han realizado investigaciones referentes a diversas condiciones para el vehículo autómata en las que afrontará, para salvaguardar la seguridad de los usuarios, a la vez se realizaban investigaciones para la reducción de contaminantes emitidos por los vehículos. Hoy en día existen normas, legislaciones y reglamentos para la implementación de estos sistemas, además se estipularon diferentes niveles de conducción autónoma conformados por la Society of Automotive Engineers (SAE)[1], en los cuales se indican a continuación:

- **Nivel 0.** La conducción no tiene nada automatizado
- **Nivel 1.** Asistencia del conductor
- **Nivel 2.** Automatización parcial
- **Nivel 3.** Automatización de conducción condicional
- **Nivel 4.** Automatización de conducción alta
- **Nivel 5.** Automatización de conducción completa

Las ventajas de los vehículos de conducción autónoma en ambientes exteriores han mejorado la seguridad vial, reducción de tráfico y optimización de tiempos. Por otra parte, las desventajas de la conducción autónoma han elevado los costos de producción, problemas con desarrollo de algoritmos eficientes, ciberseguridad.

Vinculado a los niveles de conducción autónoma, los sensores de visión son esenciales para el vehículo y la adquisición de imágenes del entorno con el que está interactuando, la información adquirida de los sensores de visión se utiliza para diferentes aplicaciones como el seguimiento de objetos y la detección de objetos, este último es el que se discutirá más a fondo al respecto.

La detección de objetos es una de muchas aplicaciones en las que se puede utilizar los sensores de visión, implementando en sistemas de conducción autónoma. Principalmente, se propone un clasificador desarrollado para detectar los objetos con que interactúa un vehículo, a través de un modelo entrenado que contiene las características necesarias para analizar y detectar cada objeto sobre el que se ha validado el modelo.

El clasificador funciona a través de una red neuronal convolucional (RNC) encargada de analizar los datos del conjunto mediante la validación y las pruebas, de modo que cuando se entrena el modelo se obtengan resultados precisos al procesar los datos y obtener una predicción de los objetos. Las RNC responsables de efectuar el procesamiento de imágenes se crearon a través de diferentes arquitecturas, dedicadas a la extracción de características de la imagen.

1.1. Antecedentes

El procesamiento tradicional de imágenes, comparado con el Aprendizaje Profundo (AP), para la detección de objetos, su principal diferencia es que el AP aprende de las características extraídas y el procesamiento tradicional

de imágenes procesa las características. Aunque el AP necesita un poder de cómputo mayor a del procesamiento tradicional de imágenes, genera una desventaja en los clasificadores y las redes neuronales pueden procesar un conjunto de datos bastante amplio [2].

La comparativa entre diferentes algoritmos de clasificación para RNC dio como resultado que GoogleNet es una de las arquitecturas adecuadas para este tipo de aplicaciones, seguida por la red VGGNet y Clarifai, también la comparación de las redes neuronales para el ángulo de rotación en caminos, demuestra que GoogleNet, es una de las redes con un muestreo balanceado en el cambio de carril mediante sus sensores de visión [3].

La detección de objetos mediante sensores de visión es un tema con el que se debe tener cuidado, ya que es necesario tener un algoritmo lo bastante entrenado para la detección de estos. Puesto que mediante lo adquirido se podría realizar la visualización por medio de datos de nube de puntos dispersos para detectar la localización de forma precisa y rápida del objeto. Para lograr esto se depende del modelo convolucional con el que se trabaje y el sensor de visión para la adquisición de estos datos [4].

La región de interés es un punto que se debe tener muy presente cuando se plantea el trabajar en un objeto en específico, ya que se unifica con RNC y la detección de objetos con la cual se planea trabajar en tiempo real para detectar el objeto [5].

Existe un método para detectar objetos 3D que combina dos herramientas, la primera utilizada es el sensor LiDAR para medir la profundidad de la imagen mostrando el ojo de pájaro y los puntos en la nube usando la imagen del conjunto de datos (dataset) proporcionada por KITTY; Para la detección de objetos se utilizó CrossFusion a través de imágenes RGB para extraer características de la imagen, logrando la combinación de estas dos herramientas desarrolladas, creando mapas y detectando objetos 3D [6].

Entre los muchos métodos que existen para la detección de objetos, la clasificación de regiones intenta recorrer los polos de la imagen para confirmar

que no hay características en estas partes de interés, y cuando esta parte se elimina, pasa al análisis de partes faltantes para descubrir y extraer características de los objetos de interés, también hay optimización de regiones como una forma de realizar análisis. El procesamiento de la información dentro del proceso de imagen a través de operaciones de color, borde y superpíxel que conducen a la localización del objeto. Finalmente, la clasificación de la región implementa diversas operaciones de AP a través de la compleja red neuronal, obteniendo así la detección de objetos [7].

1.2. Objetivos

1.2.1. Objetivo general

Diseño e implementación de módulos de visión 2D para la detección de camino y objetos que ayuden a la navegación de un vehículo autónomo. Las pruebas en este proyecto serán rutas locales con el fin de personalizar el contexto de navegación de vehículos autónomos.

1.2.2. Objetivos específicos

1. Construir una base de imágenes con diferentes clases de objetos que pueden encontrarse en un ambiente urbano durante la conducción de un vehículo en calles locales y desde bases de imágenes del estado del arte.
2. Analizar las características del camino o carretera local en secuencias de video adquiridas desde un vehículo en movimiento.
3. Programar diferentes arquitecturas basadas en aprendizaje profundo que permitan la clasificación de nuestros datos de entrada.
4. Reducir el tiempo de procesamiento del clasificador de objetos desde redes de aprendizaje profundo mediante el uso de estrategias dispersas.

5. Validar el clasificador de objetos mediante diferentes métricas y comparar con estrategias similares del estado del arte.

1.3. Justificación

La navegación de vehículos autónomos tiene propuestas que se enfocan en la construcción de autos inteligentes para uso personal, servicios de transporte público e industrial en ambientes urbanos. Se ha creado un espacio que día a día se van incorporando estos vehículos en los caminos, las reformas viales se han actualizado para el acoplamiento de estos vehículos en las ciudades. Las herramientas tecnológicas que desarrollan los sistemas autónomos para el control interno, mediante las redes inalámbricas con las que operan. Los sistemas de conducción autónoma deben tener un gran poder de cómputo para procesar toda la información que los sensores envía al sistema, manteniendo una velocidad rápida de envío y recepción de datos desde la nube [8].

Si bien este es un gran propósito, diferentes aplicaciones se pueden alcanzar en el diseño de estrategias para la navegación inteligente de vehículos. Lo más reciente, propone el diseño de robots autónomos para el servicio de ayuda en lugares cerrados ante la llegada de personas o cada cierto tiempo durante el día. Otra aplicación muy interesante desde hace tiempo ya es la navegación inteligente de robots para la asistencia de personas enfermas o de la tercera edad.

La interacción humano-computadora requiere una estrategia de navegación en ambientes interiores, sin embargo, precisa y con alta velocidad de procesamiento para lograr los objetivos funcionales. Así, muchas aplicaciones pueden surgir en el desarrollo de módulos para la conducción autónoma de vehículos, dependiendo del ambiente específico de interacción, el diseño de las arquitecturas de software y hardware son muy diversas.

Una gran cantidad de sensores integrados en el vehículo pueden causar fal-

ta de sincronización o evitar el cálculo de una tarea esencial, por ello, eso tampoco es una solución. En este trabajo se pretende utilizar únicamente cámaras, analizando estrategias de visión 2D dispersas para aligerar el tiempo de procesamiento y que puedan responder a condiciones ambientales locales.

Estas características se pondrán a prueba en modelos convolucionales para la elección de cada una de las diferentes arquitecturas existentes en las que se podrán realizar pruebas de entrenamiento, sobre las estrategias para extraer las mejores técnicas y ajustar nuestro objetivo de navegación.

1.4. Contenido del documento

En los siguientes capítulos se hablará al respecto de los conceptos importantes sobre las redes neuronales y las arquitecturas que tienen las seis redes (AlexNet, LeNet5, ResNet 50, ResNet 101, VGG16 y VGG19) que se estarán probando para la detección de objetos; También sé explicará la adquisición del conjunto de datos en ambientes urbanos al exterior y las problemáticas que se solucionaron para mejorar la adquisición de datos en ambientes reales para las 11 clases que se analizarán. Dentro del capítulo 4 se dará a conocer la red que funcionó de forma adecuada para la detección de objetos, al igual el entrenamiento y validación del modelo para cada red, demostrando por último la clasificación de objetos mediante el ingreso de imágenes, arrojando la localización del objeto y la predicción.

Capítulo 2

Estado del arte

El presente capítulo está enfocado en la descripción de las herramientas para el procesamiento de imágenes básico que se utiliza en aprendizaje profundo, además se describen las redes neuronales convolucionales (RNC) y sus diversas capas que extraen características para utilizarlas en detección de objetos.

2.1. Introducción

En la primera parte de este capítulo se dará una explicación de dos metodologías diferentes para el procesamiento de datos, las cuales son el procesamiento tradicional de imágenes (PTI) y el aprendizaje profundo, en los cuales existe una vertiente, que marca una diferencia principal en sus procesos [9].

El aprendizaje profundo es entrenado y el procesamiento tradicional de imágenes es una técnica que requiere de una etapa de procesamiento mediante técnicas tradicionales. Por lo tanto, el aprendizaje profundo brinda

la posibilidad de trabajar con una gran cantidad de datos para procesar en la RNC que aprende acorde a lo que se le ingresa, se podría decir que el aprendizaje profundo es muy similar a programación de bloques.

El procesamiento tradicional de imágenes es una técnica basada en el análisis espacial de las imágenes, se trabaja mediante umbrales, colores y bordes, sin embargo, el procesamiento tradicional de imágenes no tiene un aprendizaje de los datos ingresados. También es necesario mencionar que a diferencia del aprendizaje profundo, los procesos que realiza dentro de la programación son transparentes, y esto hace más sencillo el detectar dónde está un problema a solucionar que pueda tener el código.

En la tabla 2.1 se realiza una comparativa de las dos metodologías explicadas.

Tabla 2.1: Tabla comparativa procesamiento tradicional de imágenes vs AP [2].

Criterios	PTI	AP
Entrenamiento del conjunto de datos	Pequeño	Grande
Poder de cómputo	Bajo	Elevado
Tiempo de entrenamiento	Pequeño	Largo
Transparencia del algoritmo	Elevado	Bajo
Experiencia en el campo	Elevado	Bajo
Flexibilidad	Elevado	Bajo

2.2. Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales (RNC) se utilizan para procesar grandes cantidades de datos, por su potencia computacional, siendo una buena opción para el reconocimiento de patrones, con las cuales se pueden procesar matrices por medio de capas convolucionales, capas no lineales, capas de Agrupamiento y capas completamente conectadas [10], las cuales se explicarán a continuación.

2.2.1. Capa convolucional (Convolutional Layer)

La capa convolucional es desarrollada mediante filtros que dependen de los parámetros de entrada, los cuales son el ancho, altura y profundidad. El ancho y la altura dependen de los píxeles, y la profundidad depende del número de canales. En esta situación son los tres canales de RGB como se visualiza en la figura 2.1.

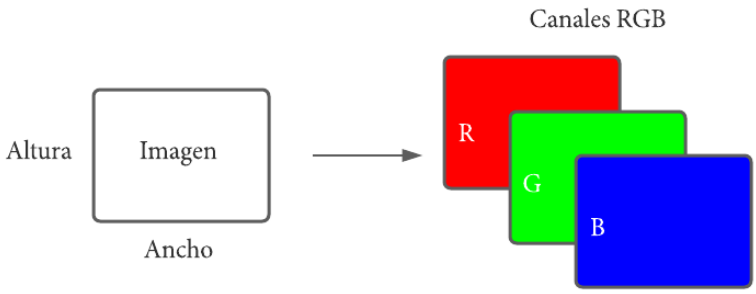

Figura 2.1: Parámetros de entrada para la capa de convolución.

Conociendo los parámetros necesarios de la imagen para la capa convolucional, prosigue el proceso de comunicación con cada filtro para analizar la altura, el ancho y la profundidad de la imagen a través del kernel de procesamiento, y las propiedades requeridas para los mapas de activación [11].

Como ejemplo se usará el del conjunto de datos CIFAR-10, que tiene imágenes de 32 x 32 x 3, lo que significa que tienen 32 de ancho, 32 de alto y 3 de profundidad (RGB). La capa de convolución requiere de un kernel de 3 x 3, lo que nos da un número de pesos de 3 x 3 x 3 = 27, hasta este punto la neurona ha aumentado la profundidad a medida que se usan más filtros y el tamaño ahora es de 16 x 16 x 94. Asimismo, al utilizar un kernel igual o de diferente tamaño, la profundidad aumenta y el tamaño de los datos de entrada disminuye [12], en la figura 2.2, se visualiza la operación realizada por la capa de convolución.

Dentro de la capa de convolución también existen tres parámetros importantes que se deben mencionar para la optimización de la convolución, los

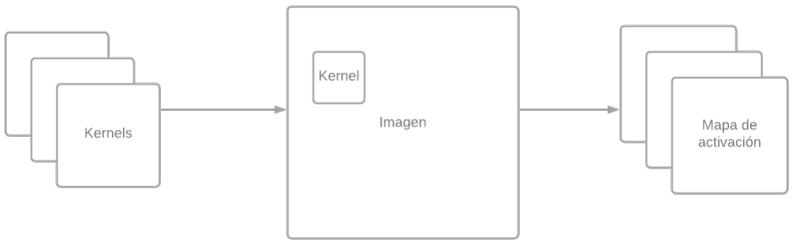

Figura 2.2: Proceso de una capa de convolución.

cuales son: la profundidad (depth), el paso (stride) y el relleno de ceros (zero-padding).

Profundidad (Depth)
La profundidad del volumen de salida generado por las capas convolucionales se puede ajustar manualmente por el número de neuronas en la capa en la misma región de entrada. Esto se puede ver con otras formas de Redes Neuronales Artificiales (RNA), donde todas las neuronas en la capa están conectadas directamente a cada neurona desde el frente. Esta reducción puede reducir significativamente el número total de neuronas en la red y la capacidad de reconocimiento de patrones [13].

Pasos (Stride)
También podemos definir los pasos que fijamos en la profundidad alrededor del tamaño espacial de entrada para la ubicación del campo receptivo. Por ejemplo, si establecemos un paso en 1, obtenemos un campo receptivo muy superpuesto que producirá matrices de datos grandes. De lo contrario, establecerlo en un número mayor reducirá la cantidad de interferencia y producirá una salida con un tamaño espacial más pequeño [13].

Relleno de ceros (Zero-padding)
Es un proceso simple de relleno del borde y es una forma efectiva de proporcionar un mayor control sobre el volumen de salida [13]. Es importante comprender que con estas técnicas cambiamos el tamaño espacial de la salida

de la capa de convolución. Para calcular esto se utiliza la ecuación 2.1

$$\frac{(V - R) + 2Z}{S + 1} \tag{2.1}$$

donde:

V = Altura x Ancho x Profundidad
R = Tamaño del campo receptivo
Z = Cantidad de Zero Padding
S = Stride

Capa de Activación (Activation Layer)
Una función de activación hace que la red se comporte de diferentes formas con la finalidad de extraer información compleja de los datos y representar funciones entre la entrada y la salida. Por lo tanto, al agregar no linealidad mediante funciones de activación, pueden realizar tareas desde la entrada hasta la salida. Una característica importante de la función de activación es que debe quedar clara para que podamos implementar la nueva estrategia de optimización de propagación para tener en cuenta errores o pérdidas [11].

Tipos de funciones de activación

- Función de escalón binario (Binary Step Function)
- Lineal (Linear)
- Sigmoidea (Sigmoid)
- Tangente Hiperbólica (Tanh)
- Unidad Lineal Rectificada (ReLU)

Escalón binario
La función de activación más simple que existe y se puede implementar con

declaraciones simples como se muestra en la figura 2.3 y se representa mediante la ecuación 2.2. Sin embargo, la función de conversión binaria no se puede utilizar en el caso de clasificación de múltiples clases [14]. Además, el gradiente de la función de orden binario es cero, lo que puede dificultar el proceso de propagación, si se calcula la derivada de $f(x)$ con respecto a x, es cero como:

$$f(x) = \begin{cases} 1 & \text{si} \quad x \geq 0 \\ 0 & \text{si} \quad x < 0 \end{cases} \tag{2.2}$$

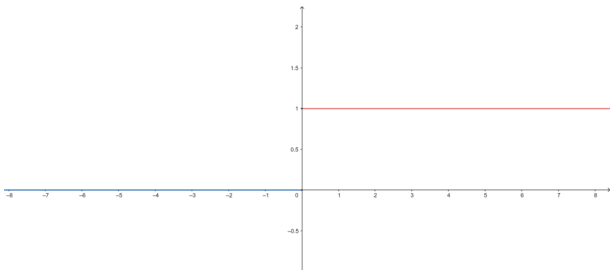

Figura 2.3: Gráfica de la función de Escalón binario.

Lineal

La función Lineal es proporcional a la entrada y se puede definir como:

$$f(x) = ax \tag{2.3}$$

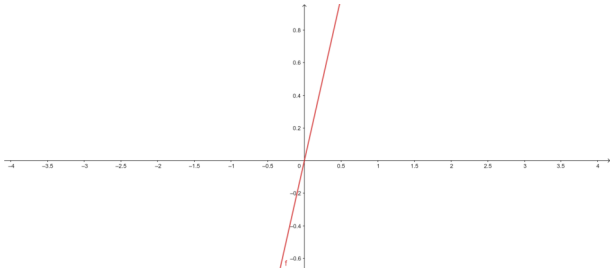

Figura 2.4: Gráfica de la función Lineal, para $a = 2$.

El valor de la variable a puede ser cualquier valor, la derivada de la función $f(x)$ y el gradiente son igual a, el beneficio al usar la función no optimizará el error porque el valor del gradiente es el mismo para cada iteración, además la red no podrá identificar patrones complejos a partir de los datos. Por lo tanto, las funciones lineales son ideales cuando se requiere tareas simples [14], y se muestra en la figura 2.4 y se representa mediante la ecuación 2.3.

Sigmoidea

La función Sigmoidea es la más utilizada porque no es lineal como se muestra en la figura 2.5 y se representa mediante la ecuación 2.4, transforma valores entre 0 y 1:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.4}$$

La derivada de $f(x)$ es $1 - sigmoid(x)$, además, la función sigmoidea no es simétrica alrededor de cero, lo que significa que los signos de todos los valores resultantes para las neuronas serán los mismos [14].

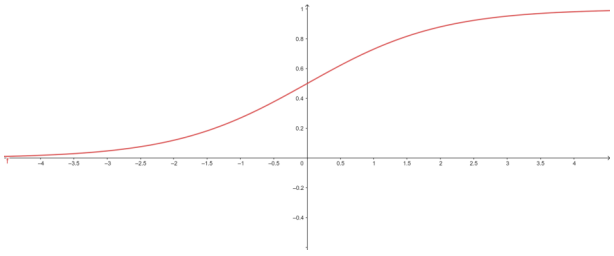

Figura 2.5: Gráfica de la función Sigmoidea.

Tangente Hiperbólica

La función Tangente Hiperbólica es simétrica alrededor del origen a comparación de la función sigmoidea. Esto da como resultado que diferentes señales de salida de las capas anteriores se pasen como entradas a la siguiente capa como se muestra en la figura 2.6 y se representa mediante la ecuación 2.5:

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{2.5}$$

La función tangente hiperbólica es continua y puede derivarse con valores que oscilan entre -1 y 1. Se prefiere la tangente hiperbólica a la función sigmoidea porque tiene gradientes que no se limitan a cambiar en una dirección particular [14].

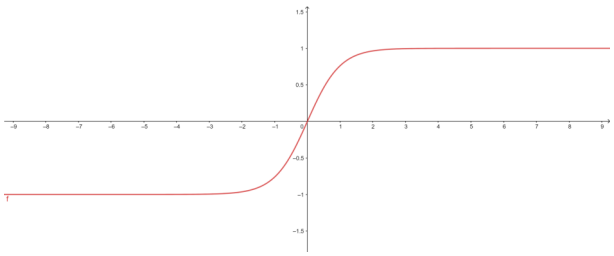

Figura 2.6: Gráfica de la función Tanh.

ReLU

La función ReLU son las siglas de Rectified Liner Unit y es una función de activación no lineal que se implementa en redes neuronales. La ventaja de utilizar la función ReLU es que no todas las neuronas están activas simultáneamente [14]. Esto significa que la neurona no se apagará hasta que la salida de la derivación lineal sea cero, como se muestra en la figura 2.7 y se representa mediante la ecuación 2.6:

$$f(x) = \begin{cases} x & \text{si} \quad x \geq 0 \\ 0 & \text{si} \quad x < 0 \end{cases} \tag{2.6}$$

2.2.2. Capa Completamente Conectada (Fully-Connected Layer)

La capa FC está completamente conectada a todas las activaciones de las capas anteriores, estas capas son las últimas en las arquitecturas de RNC. Es

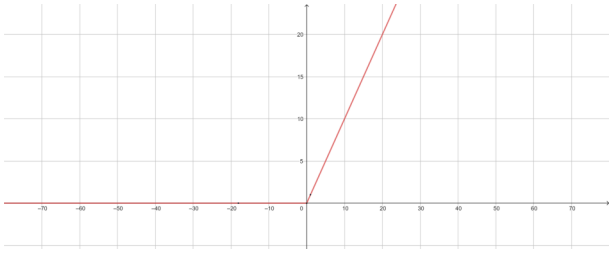

Figura 2.7: Gráfica de la función ReLU.

una práctica común usar una o dos FC antes de implementar un clasificador softmax, como lo demuestra la siguiente arquitectura (simplificada).

En la arquitectura de la figura 2.8 se utilizaron dos capas FC antes del clasificador softmax, que calculará la probabilidad de salida final para cada clase.

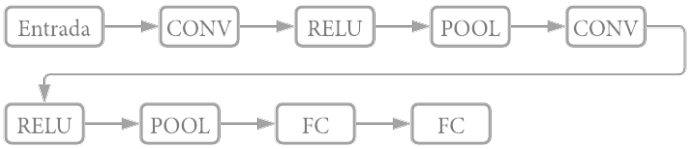

Figura 2.8: Arquitectura demostrativa de las capas FC [11].

2.2.3. Capa de agrupamiento (Pooling layer)

La capa de agrupamiento, que aquí será llamada como POOL, tiene el objetivo de reducir gradualmente el tamaño del campo espacial (altura y ancho) de entrada. Esto nos permite reducir el número de cálculos en la red, esta capa funciona de forma independiente en cada profundidad de entrada, utilizan el agrupamiento máximo, generalmente se realiza en medio de una arquitectura RNC para reducir el tamaño espacial, mientras que el agrupamiento promedio se usa a menudo como una capa de red inferior donde

queremos evitar el uso de capas FC por completo.

Normalmente, se usa un tamaño de POOL de 2 × 2 como se muestra en la figura 2.9, para las RNC más profundas con imágenes de entrada mayores a 200 píxeles. Se puede usar un tamaño de POOL de 3 × 3 al inicio de la RNC [11] en la tabla 2.2 se muestra cómo son los datos de entrada y salida al pasar por una capa de agrupamiento donde:

An_{ent} = Ancho de la imagen de entrada
Al_{ent} = Altura de la imagen de entrada
P_{ent} = Profundidad de la imagen de entrada
F = Tamaño de agrupamiento
S = Paso
An_{sal} = Ancho de la imagen de salida
Al_{sal} = Altura de la imagen de salida
P_{sal} = Profundidad de la imagen de salida

Se muestran dos tipos de ejemplo de cómo se puede utilizar el tamaño de agrupamiento y el tamaño de paso para la RNC:

Tipo 1: F = 3; S = 2 se conoce como agrupación traslapada y generalmente se aplica a imágenes / volúmenes de gran tamaño espacial.
Tipo 2: F = 2; S = 2, llamado agrupación no superpuesta. Este es el tipo más común y se aplica a imágenes con dimensiones espaciales pequeñas. Para las arquitecturas de red que aceptan imágenes de entrada pequeñas (del orden de 32 a 64 píxeles).

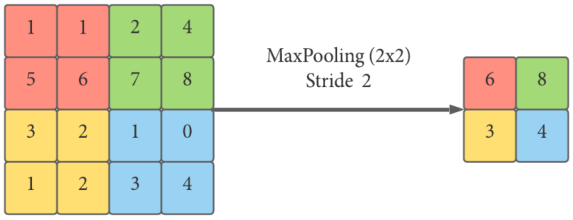

Figura 2.9: MaxPooling (2x2) y Stride = 2, representando al tipo 2 [10].

Tabla 2.2: Capa de agrupamiento.

Tipo	Tamaño	Parámetros
Entrada	An_{ent} x Al_{ent} x P_{ent}	Tamaño del campo receptivo F El Stride S.
Salida	An_{sal} x Al_{sal} x P_{sal}	$An_{Sal} = ((An_{en} - F)/S) + 1$ $Al_{sal} = ((Al_{ent} - F)/S) + 1$ $P_{sal} = P_{ent}$

2.2.4. Capa de Normalización por Lotes (Batch Normalization Layer)

La Normalización por lotes, es un método para coordinar la actualización de varias capas en el modelo de una manera fácil de restablecer casi cualquier red profunda. Se hace escalando la salida de la capa, incluida la normalización de las activaciones de cada variable de entrada para cada mini lote, la activación de un nodo de la capa anterior como se muestra en la figura 2.10. La normalización se refiere a medir los datos para que tengan una media de 0 y una desviación estándar de 1. Este proceso también se conoce como blanqueamiento cuando se aplica a imágenes de visión por computadora. Al blanquear las entradas de cada categoría, damos un paso hacia el logro de distribuciones de entrada estables que eliminan los efectos de la covariable interna.

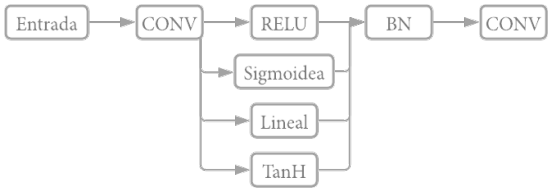

Figura 2.10: Capa de normalización por lotes [10].

Normalizar los factores significa que el alcance y la distribución de las entradas durante la actualización del peso no cambiarán, al menos no de manera significativa. Esto tiene el efecto de estabilizar y acelerar el entrenamiento

de la red neuronal profunda.

La normalización de la entrada de la clase tiene un efecto en el entrenamiento del modelo, lo que reduce en gran medida el número de épocas necesarias. También puede tener el mismo efecto de regularización, reduciendo errores. La normalización por lotes puede tener un impacto significativo en el rendimiento de la optimización, especialmente para redes complejas y redes no lineales.

La normalización tiene un efecto fundamental en la configuración de la red, hace que el contexto del problema de optimización relacionado sea significativamente más fluido. En particular, esto asegura que los gradientes sean más predictivos y, por lo tanto, permite el uso de una gama más amplia para las tasas de aprendizaje y una convergencia de la red más rápida [15].

2.2.5. Capa de Abandono (Dropout Layer)

La capa de abandono es una técnica que evita el sobreentrenamiento y proporciona una manera de combinar de manera eficiente muchas arquitecturas de redes neuronales diferentes de manera casi exponencial. El término "dropout"se refiere al descarte de unidades en una red neuronal. Al borrar una unidad, se refiere a que se eliminó temporalmente de la red, junto con todas sus conexiones entrantes y salientes, como se visualiza en la figura 2.11 la figura izquierda se muestra una RNC y a la derecha se muestra una RNC con capas de abandono.

La selección de la unidad que se va a borrar es aleatoria. En el caso más simple, cada unidad se mantiene con una probabilidad constante independiente de la otra, ya que se puede elegir usando un verificador o simplemente se puede establecer en 0.5, y parece estar cerca del óptimo para muchos tipos de redes. Sin embargo, para las unidades de entrada, la probabilidad de retención óptima suele estar más cerca de 1 que de 0.5 [16].

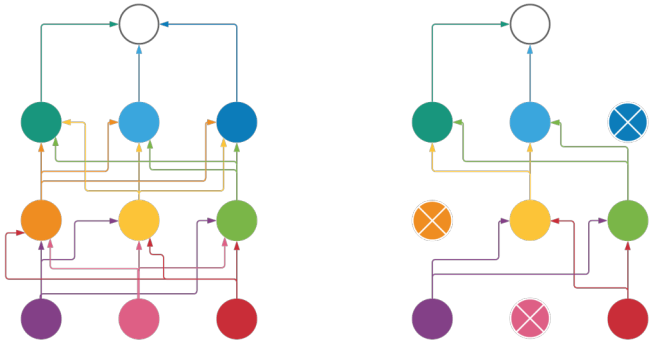

Figura 2.11: Aplicación de la capa de abandono.

2.3. Arquitecturas de RNC

En los últimos años, se han logrado grandes avances en este campo de visión por computadora y las redes que se mostrarán son significativas, ya que fueron de las principales redes participantes en la competencia ImageNet Large Scale Visual Recognition Challenge (ILSVRC), las cuales han logrado tener buenos resultados ante dicha competición. Una de las técnicas que utilizán para mejorar el rendimiento de las RNC es el incrementar el tamaño de la red [17]. En esta sección se hablará a detalle sobre las arquitecturas y sus características.

2.3.1. AlexNet

Es una arquitectura desarrollada basándose en ocho capas, las cuales cinco capas son convolucionales y tres capas son completamente conectadas, permitiendo a la red ser capaz de reconocer objetos con un menor enfoque. Las características principales de la RNC AlexNet se describen en la tabla 2.3. La red obtiene más información de fondo no relacionada a su última capa convolucional, lo que en ocasiones afecta en el clasificador al predecir el objeto [18].

Tabla 2.3: Características AlexNet [19].

Característica	Descripción	Parámetros
Función de activación	Poco tiempo de entrenamiento	ReLU
Poder de cómputo	Requiere de tarjeta gráfica	GPU > 3 GB
Capa de agrupamiento	Reducción del error a 0.5 %	Superposición
Capa de abandono	Mayor tiempo de entrenamiento	50 %
ILSVRC	Ganador con tasa de error 15.3 %	Año 2012

Por último se demuestra la arquitectura, las capas, activaciones y tamaños de cada una de las capas que contiene AlexNet en la tabla 2.4, comenzando por la capa de entrada del conjunto de imágenes, se define como una capa que filtra las dimensiones de la imagen (227 x 227 x 3). Para la primera capa convolucional se utiliza una profundidad de 96, para la segunda capa convolucional se utiliza una profundidad 256 y en la tercera capa convolucional se utiliza una profundidad de 384.

Tabla 2.4: Arquitectura AlexNet.

Capa	Tamaño	Tamaño del kernel	Pasos
Entrada	224 x 244 x 3		
Conv. (ReLU)	55 x 55 x 96	11 x 11	4
Agrupamiento Máx.	27 x 27 x 96	3 x 3	2
Conv. (ReLU)	27 x 27 x 256	5 x 5	1
Agrupamiento Máx.	13 x 13 x 256	3 x 3	2
Conv. (ReLU)	13 x 13 x 384	3 x 3	1
Conv. (ReLU)	13 x 13 x 384	3 x 3	1
Conv. (ReLU)	13 x 13 x 256	3 x 3	1
Agrupamiento Máx.	6 x 6 x 256	3 x 3	2
Comple. Conectada	9216		
Comple. Conectada	4096		
Comple. Conectada	4096		
Salida	1000		

Las capas convolucionales 1, 2 y 5 operan en conjunto con una capa de agrupamiento máximo (3 x 3) y con paso de 2. La conexión con 3 capas, la primera con los parámetros de salida de las convoluciones y dos capas completamente conectadas por 4096 nodos. Para finalizar la RNC se realiza

la activación de Softmax para determinar la probabilidad de clasificación empleada por la salida final del clasificador, esta probabilidad de clasificación en la categoría Softmax puede generar hasta 1000 categorías [20].

2.3.2. ResNet

Es una arquitectura residual que combina redes residuales y convolucionales, utilizan bloques residuales, conservan el beneficio de optimizar las conexiones abreviadas de identidad al tiempo que mejoran la expresividad y la facilidad para eliminar información innecesaria. Las ResNet son arquitecturas no lineales y de forma exponencial, basándose en un estado de flujo paralelo formando filtros convolucionales.

Tabla 2.5: Arquitecturas ResNet [21]

Capa	Tamaño	18	34	50	101	152
Conv.	112 x 112	7 x 7, 64 Pasos de 2				
Conv.	56 x 56	3 x 3 Agrupamiento Máx., Pasos de 2				
		3 x 3,64 3 x 3,64 R = 2	3 x 3,64 3 x 3,64 R = 3	1 x 1,64 3 x 3,64 1 x 1,64 R = 3	1 x 1,64 3 x 3,64 1 x 1,64 R = 3	1 x 1,64 3 x 3,64 1 x 1,64 R = 3
Conv.	28 x 28	3 x 3,128 3 x 3,128 R = 2	3 x 3,128 3 x 3,128 R = 4	1 x 1,128 3 x 3,128 1 x 1,512 R = 4	1 x 1,128 3 x 3,128 1 x 1,512 R = 4	1 x 1,128 3 x 3,128 1 x 1,512 R = 8
Conv.	14 x 14	3 x 3,256 3 x 3,256 R = 2	3 x 3,256 3 x 3,256 R = 6	1 x 1,256 3 x 3,256 1 x 1,1024 R = 6	1 x 1,256 3 x 3,256 1 x 1,1024 R = 23	1 x 1,256 3 x 3,256 1 x 1,1024 R = 36
Conv.	7 x 7	3 x 3,512 3 x 3,512 R = 2	3 x 3,512 3 x 3,512 R = 3	1 x 1,512 3 x 3,512 1 x 1,2048 R = 3	1 x 1,512 3 x 3,512 1 x 1,2048 R = 3	1 x 1,512 3 x 3,512 1 x 1,2048 R = 3
Salida	1 x 1	Agrupamiento Prom., 1000, Comple. Conectada, Softmax				

Cuenta con conexiones directas, beneficiando el flujo residual, entre cada

conjunto de procesamiento, permitiendo olvidar los datos del estado anterior, esta red ha dejado atrás a otras RNC dando mejores resultados para la detección de objetos [22]. También fue ganadora en 2015 de la competencia ILSVRC. Dependiendo del número de capas que tenga la red residual su error se va disminuyendo proporcionalmente respecto al incremento de capas, tomando en cuenta que existen diferentes tamaños, en la tabla 2.5 se explica la arquitectura de las cinco dimensiones de ResNet diferentes, donde R es igual al número de iteraciones.

2.3.3. VGGNET

La red VGGNET, obtuvo el segundo lugar en la competencia ILSVRC del 2014, esta red realiza un procesamiento que resta el valor promedio de los tres canales RGB de cada píxel. La imagen a procesar pasa por diferentes conjuntos de capas convolucionales, con filtros para un campo receptivo pequeño de dimensiones de 3×3, las cuales ayudan a procesar mejor la imagen de izquierda, derecha, arriba, abajo y centro del filtro.

También se utilizan filtros de convolución de 1×1, que se denotan como una transformación lineal de los canales de entrada. Los conjuntos de capas convolucionales se conectan a tres capas totalmente conectadas, las primeras dos capas están formadas por 4096 canales, la tercera tiene 1000 canales (uno para cada clase) y por último está la capa softmax [23].

Las diferentes redes de VGGNET, se describen en la tabla 2.6, cada VGG tiene la misma forma de la que se habla en el párrafo anterior y son diferentes sólo en la profundidad.

Tabla 2.6: Arquitecturas VGGNET [23]

Redes VGGNET					
A	A-LRN	B	C	D	E
11 Capas	11 Capas	13 Capas	16 Capas	16 Capas	19 Capas
Entrada: 224 x 224 x 3					
Conv. 3 x 3 x 64	Conv. (LRN) 3 x 3 x 64	Conv. 3 x 3 x 64 3 x 3 x 64	Conv. 3 x 3 x 64 3 x 3 x 64	Conv. 3 x 3 x 64 3 x 3 x 64	Conv. 3 x 3 x 64 3 x 3 x 64
Agrupamiento Máximo					
Conv. 3 x 3 x 128	Conv. 3 x 3 x 128	Conv. 3 x 3 x 128 3 x 3 x 128	Conv. 3 x 3 x 128 3 x 3 x 128	Conv. 3 x 3 x 128 3 x 3 x 128	Conv. 3 x 3 x 128 3 x 3 x 128
Agrupamiento Máximo					
CConv. 3 x 3 x 256 3 x 3 x 256	Conv. 3 x 3 x 256 3 x 3 x 256	Conv. 3 x 3 x 256 3 x 3 x 256	Conv. 3 x 3 x 256 3 x 3 x 256 1 x 1 x 256	Conv. 3 x 3 x 256 3 x 3 x 256 3 x 3 x 256	Conv. 3 x 3 x 256 3 x 3 x 256 3 x 3 x 256 3 x 3 x 256
Agrupamiento Máximo					
Conv. 3 x 3 x 512 3 x 3 x 512	Conv. 3 x 3 x 512 3 x 3 x 512	Conv. 3 x 3 x 512 3 x 3 x 512	Conv. 3 x 3 x 512 3 x 3 x 512 1 x 1 x 512	Conv. 3 x 3 x 512 3 x 3 x 512 3 x 3 x 512	Conv. 3 x 3 x 512 3 x 3 x 512 3 x 3 x 512 3 x 3 x 512
Agrupamiento Máximo					
Conv. 3 x 3 x 512 3 x 3 x 512	Conv. 3 x 3 x 512 3 x 3 x 512	Conv. 3 x 3 x 512 3 x 3 x 512	Conv. 3 x 3 x 512 3 x 3 x 512 1 x 1 x 512	Conv. 3 x 3 x 512 3 x 3 x 512 3 x 3 x 512	Conv. 3 x 3 x 512 3 x 3 x 512 3 x 3 x 512 3 x 3 x 512
Agrupamiento Máx.					
Comple. Conectada - 4096					
Comple. Conectada - 4096					
Comple. Conectada - 1000					
Softmax					

2.3.4. LeNet5

Es una RNC con suficientes entradas para contener muchos objetos y múltiples salidas, lo que le permite ser relacionada como una red neuronal de desplazamiento espacial (SDNN), que es capaz de reconocer secuencias en una sola transmisión sin segmentación previa, como se muestra en la tabla 2.7. Las clases de escasa complejidad y los clústeres extremos son el kernel de la familia de modelos LeNet5, las capas inferiores incluyen capas convolucionales y de agrupamiento máximo, las capas superiores están completamente conectadas. La entrada a la primera capa completamente conectada es el conjunto de mapas de características de la capa inferior[24].

Tabla 2.7: Arquitectura LeNet5 [25]

Capa	Tamaño	Tamaño del kernel	Pasos
Entrada	32 x 32 x 3		
Conv. (Tanh)	28 x 28 x 6	5 x 5	1
Agrupamiento Prom.	14 x 14 x 6	2 x 2	2
Conv. (Tanh)	10 x 10 x 16	5 x 5	1
Agrupamiento Prom.	5 x 5 x 16	2 x 2	2
Conv. (Tanh)	120	5 x 5	1
Comple. Conectada	84		
Comple. Conectada	10		
Salida (Softmax)	10		

2.3.5. GoogleNet

La red GoogleNet logró obtener el primer lugar en la competencia ILSVRC del 2014, esta red está basada en un estudio de Inception para evaluar los resultados del algoritmo, esta red utiliza componentes densos. Con algunas modificaciones, la brecha se amplía y el inicio es especialmente útil en el contexto de la detección y localización de objetos. La estructura de Inception es una red neuronal convolucional, puede aproximarse y cubrirse con componentes, bloques convolucionales los cuales se construyen capa por ca-

pa y se analizan las estadísticas de la última capa que pasa por bloques de agrupamiento, formando unidades de la capa posterior conectando con la unidad de la capa anterior. Además, las capas de Inception son repetitivas, lo que da como resultado un modelo de profundidad de 22 capas en el caso del modelo GoogLeNet. Dentro del desarrollo de GoogleNet y como se visualiza en la tabla 2.8 tiene capas convolucionales que se activan mediante la función ReLU; La dimensión de su campo receptivo en la red es 224 x 224 x 3 con diferentes reducciones de 3 x 3 y 5 x 5, también utiliza filtros de 1 x 1 en la capa de proyección después de la capa de agrupamiento máximo. La red tiene un total de 22 capas o 27 capas contando las capas de agrupamiento, la implementación de la agrupamiento promedio ayuda al clasificador mediante una capa lineal para facilitar el análisis de las clases [17].

Tabla 2.8: Arquitectura GoogleNet [17]

Capa	Tamaño	Tamaño del kernel	Pasos
Conv.	112 x 112 x 64	7 x 7	2
Agrupamiento Máx.	56 x 56 x 64	3 x 3	2
Conv.	56 x 56 x 192	3 x 3	1
Agrupamiento Máx.	28 x 28 x 192	3 x 3	2
Inception (3a)	28 x 28 x 256		
Inception (3b)	28 x 28 x 480		
Agrupamiento Máx.	14 x 14 x 480	3 x 3	2
Inception (4a)	14 x 14 x 512		
Inception (4b)	14 x 14 x 512		
Inception (4c)	14 x 14 x 512		
Inception (4d)	14 x 14 x 528		
Inception (4e)	14 x 14 x 832		
Agrupamiento Máx.	7 x 7 x 832	3 x 3	2
Inception (5a)	7 x 7 x 832		
Inception (5b)	7 x 7 x 1024		
Agrupamiento Prom.	1 x 1 x 1024	7 x 7	1
Abandono (40 %)	1 x 1 x 1024		
Lineal	1 x 1 x 1000		
Softmax	1 x 1 x 1000		

2.4. Arquitecturas R-RNC

Las redes neuronales convolucionales basadas en regiones (R-RNC) son bastante buenas en analizar pequeñas regiones de la imagen, haciendo que sean la mejor opción para detectar objetos diminutos en imágenes de gran tamaño tomando pequeñas regiones de la imagen en un rango de 2,000 regiones a analizar, también es una alternativa para la localización de objetos. La implementación de una R-RNC se describe mediante 3 pasos como se visualiza en la figura 2.12; Primero se toma el conjunto de datos de entrada de acorde a las características que necesita tener para ser procesada la imagen; El segundo paso es el extraer las 2,000 regiones propuestas para procesar. El tercer paso es el desarrollar una RNC para el análisis y extracción de las características que ya se vieron en la sección de redes neuronales convolucionales [26].

Figura 2.12: R-RNC.

2.4.1. R-RNC acelerada

Este tipo de red trabaja mejor con una tarjeta gráfica (GPU) a comparación de otras que trabajan con la CPU de la computadora. Ya que el procesamiento de los datos de entrada y los procesos que realizan son bastante pesados y con una GPU brinda el beneficio de procesar los de datos en un menor tiempo [27].

La red R-RNC acelerada, como se puede ver en la figura 2.13, utiliza una imagen del conjunto de datos y una sección de la región propuesta, primero

realiza un proceso mediante una RNC, después cada propuesta pasa por una capa de agrupamiento para la extracción de las características, después pasa a una capa completamente conectada de la cual resultan dos salidas. La primera salida genera una aproximación de probabilidad sobre la clase del objeto a clasificar y la segunda salida genera cuatro capas que codifica posiciones del cuadro delimitador para una de las clases. El cuadro está formado por 4 características, las cuales son altura, ancho y dos coordenadas de posición del extremo superior izquierdo [28].

Figura 2.13: R-RNC acelerada.

2.4.2. Máscara R-RNC

Este tipo de red es muy similar a la R-RNC acelerada, trabaja mejor con una GPU y procesan los datos más rápido que si se usara la CPU.

La única diferencia que tiene a comparación de la red anterior que es la máscara R-RNC es que se agrega una tercera salida como se visualiza en la figura 2.14 la cual segmenta al objeto de interés, asignado mediante una máscara de color la localización espacial más exacta dejando de un lado sólo las etiquetas y la delimitación mediante cuadros.

La máscara realiza un procesamiento binario respecto a la región de interés, para continuar analizando las siguientes regiones extraídas, teniendo una máscara diferente para cada objeto, en el caso de un clasificador sería una máscara para cada clase a reconocer [29].

Figura 2.14: Máscara R-RNC.

2.5. Ambientes interiores y exteriores

El problema de la visualización de la imagen en escenarios interiores y exteriores ha sido un desafío en el campo de visión por computadora, todo esto se debe a la iluminación de la imagen y cómo la percibe el sensor de visión.

En investigación, dentro de la adquisición de imágenes de bajo nivel se tiene el problema de la segmentación del objeto de interés en la escena. Normalmente, se utilizan características diferentes para esta metodología, las cuales son análisis por separación de canales (RGB), textura e información de frecuencia, estas características se extraen en sub-bloques de la imagen [30]. La ubicación de la imagen es una función que sólo proporciona información sobre un objeto o área específica en toda la imagen. Las características globales proporcionan una representación global de la imagen al tratarla como una entidad única, en lugar de dividir la imagen en múltiples sub-bloques y calcular las características locales de cada sub-bloque. Si bien las características globales se utilizan cada vez más para problemas de clasificación de escenas, no se han aplicado ampliamente a problemas de clasificación de objetos en interior y exterior [30].

En el estudio de clasificación de objetos en interiores y exteriores se han implementado diversas metodologías. Los histogramas se utiliza en el es-

pacio de color Ohta y coeficiente invariante de desplazamiento (DCT) al cambio después de dividir la imagen en sub-bloques analizados [31].

Se utiliza la metodología de características de color y textura de la imagen con un nivel bajo y enfoca para la localización de regiones como el césped o el cielo para brindar un apoyo en la clasificación [32].

Otra de las metodologías que se utiliza es la de máquinas de vectores de soporte (SVM) a momentos de color espacial (CM) y características de los histogramas que detectan los bordes para obtener una exactitud por encima del 80 % [33].

Por último, existe la metodología que utiliza el color de los histogramas, las características de textura y la textura basada en DCT, el cual obtiene una exactitud por encima del 85 % [34].

Capítulo 3

Metodología

En el anterior capítulo se describieron arquitecturas como AlexNet, ResNet, VGGNET, LeNet5 y GoogleNet dentro de las RNC. Se describieron dos arquitecturas de regiones para redes neuronales convolucionales como R-RNC acelerada y Máscara R-RNC. También se describe el conjunto de imágenes, mientras se conducía en un entorno local, las cuales se utilizaron para entrenar los modelos y predecir la detección de objetos. El presente capítulo está enfocado a la metodología usada dentro de las diferentes redes neuronales convolucionales (RNC) para la detección de objetos.

3.1. Diagrama General

En esta sección se ha planteado en este proyecto de tesis realiza la detección de objetos. La figura 3.1 muestra el diagrama general de la metodología. Se da inicio al proceso con una base de datos que contiene diferentes imágenes de cada una de las 11 clases, siguiendo con la implementación de la red neuronal convolucional (RNC) para el entrenamiento, al cual ingresa el 70 % de la base de datos para algunas arquitecturas que se explicaron en el

capítulo 2. Teniendo la RNC entrenada y efectuando el proceso de validación se procede a desfragmentar en parches la imagen para continuar con la detección y aproximación porcentual de cada imagen respecto a la clase a la que corresponde.

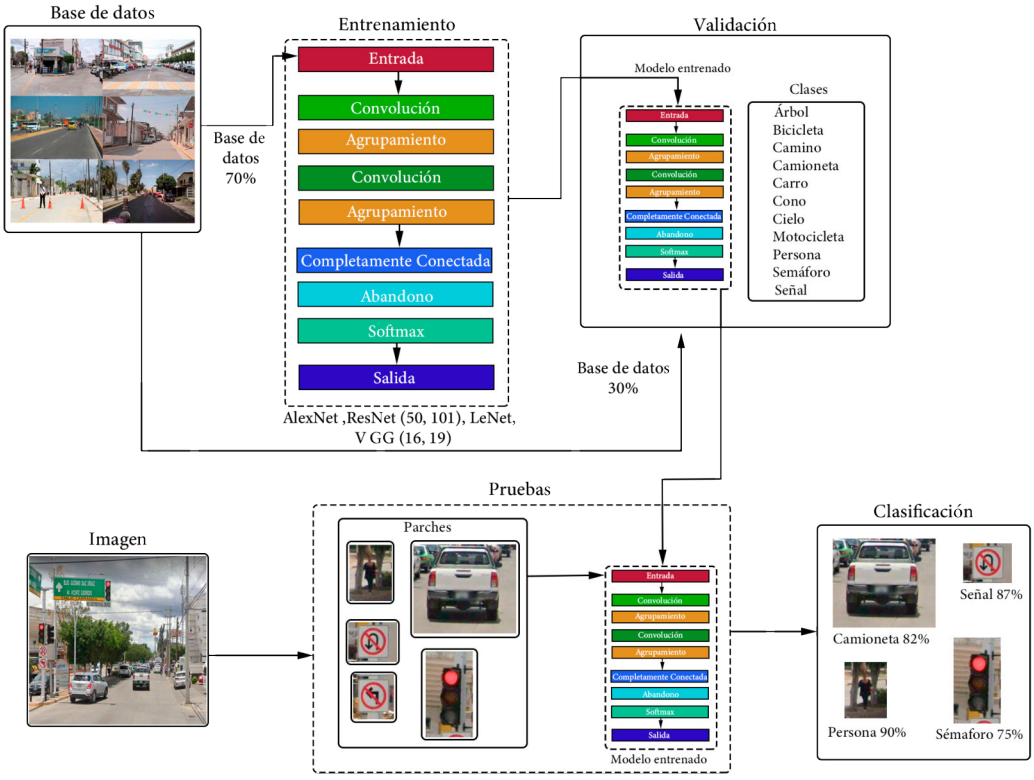

Figura 3.1: Diagrama general del sistema para clasificación de objetos basado en redes convolucionales.

3.2. Conjunto de datos (Dataset)

En la siguiente sección se hablará al respecto de cómo se formó el conjunto de datos, estableciendo diferentes aspectos para cumplir con un conjunto adecuado. En relación con el punto anterior se describen las clases con las que se trabaja a lo largo del proyecto y el motivo por el que se eligen las siguientes clases:

1. Árbol
2. Bicicleta
3. Camino
4. Camioneta
5. Carro
6. Cielo
7. Cono
8. Motocicleta
9. Persona
10. Semáforo
11. Señal (Tránsito urbano)

En cuanto a las once clases, se propusieron tres objetos con los que un vehículo tiene interacción en las rutas por las que pasa. Además de que la detección de estos objetos en ocasiones puede confundir a la RNC y poner en riesgo a los peatones.

Acerca del punto anterior, la clase árbol, bicicleta y cono se consideran en la hipótesis en que los peatones usen un vestuario similar a los colores de los árboles o conos. También se sugiere el caso donde el vehículo no perciba a una persona sobre la bicicleta, incrementando el riesgo de ocasionar un accidente.

Al mismo tiempo se recomienda el caso de la clase carro, camioneta y motocicleta, son objetos con los que se encuentra la mayor parte de tiempo sobre las rutas urbanas, a pesar de que se pudo tomar en cuenta a la clase

camión, y otras clases referentes a este tipo de vehículos, sólo se dejaron las más significativas y que cumplan con el objetivo de detectar vehículos motorizados.

Por último, en el caso de la clase señal y semáforo se debe agregar que son objetos que previenen o advierten al vehículo acerca de las condiciones o limitaciones que tiene la ruta por la que está pasando. Un ejemplo sobre la detección de semáforos, el cual tiene tres diferentes indicadores que limitan al vehículo sobre la acción a realizar, a su vez se plantea el ejemplo de los señalamientos viales, los cuales indica al vehículo sobre las características de la ruta del vehículo.

Considerando que se comprendió el porqué se eligieron las once clases, se procede a continuar con la explicación y solución aplicada para la adquisición del conjunto de datos.

3.2.1. Conjunto de datos 1

En relación con la adquisición de datos, se propuso descargar la mayor parte de las imágenes de internet, de la cuales se obtuvieron 1,000 imágenes por cada clase, y se declaró un total de 11,000 imágenes que formarían el conjunto de datos. Teniendo en cuenta que esta técnica tendría varios factores que afectarían en el entrenamiento para la detección de los objetos.

En relación con el entrenamiento se planteó la hipótesis de que el tamaño de los objetos en las imágenes variaba y esto afectaría en la calidad de los objetos ubicados en el fondo de la imagen. Otro aspecto importante en cuanto a la diversidad de clases que contiene una imagen y puede llegar a extraer características innecesarias de objetos que se sitúan en el fondo, por lo tanto, cuando se entrenará y se validará podría generar falsos-positivos en los resultados para la detección de objetos.

En consecuencia, con esos pequeños detalles significativos se optó por reali-

zar un segundo conjunto de datos con otra metodología que simplifique los aspectos que no favorecían a la RNC.

3.2.2. Conjunto de datos 2

En cuanto al segundo conjunto de datos se planteó una dinámica diferente, a causa de descargar las imágenes de internet se decidió el capturar las imágenes en un entorno real, mediante una cámara de video, cumpliendo con la adquisición de datos. Este dispositivo se instaló dentro de la cabina del conductor en el centro del tablero de una camioneta tipo pick-up.

Con respecto al tiempo de grabación se estableció que el panorama no estuviera nublado, también se tomó en cuenta el horario de 15:00 a 17:00 horas para que el brillo del sol no afectara en el umbral del video y la imagen no fuera afectada percibiendo mejor el objeto en la grabación.

Por otra parte, se grabaron 10 videos diferentes, la duración total de los videos fue de una hora y media, completando un recorrido total de 11.5 km dentro de la ciudad de Irapuato, Gto., el recorrido realizado fue desde el norte hasta el centro de la ciudad y del centro al norte de regreso, en una ruta urbana.

La técnica que se implementó fue la extracción de los frames, usando la separación de 29 Frames por segundo (FPS) para una captura del video y proceder a realizar los parches necesarios como se muestra en la figura 3.2. En vista de la problemática de segmentar y extraer los objetos de cada frame, se procedió a implementar una máscara R-RNC y la librería de PixelLib[35], con un modelo entrenado de COCO [36], este proceso simplifica la extracción de objetos, dando como resultado el objeto con un fondo limpio, además que al finalizar este proceso se notó un desequilibrio en la mayoría de las clases, por lo que se efectuó la misma dinámica en una ciudad diferente.

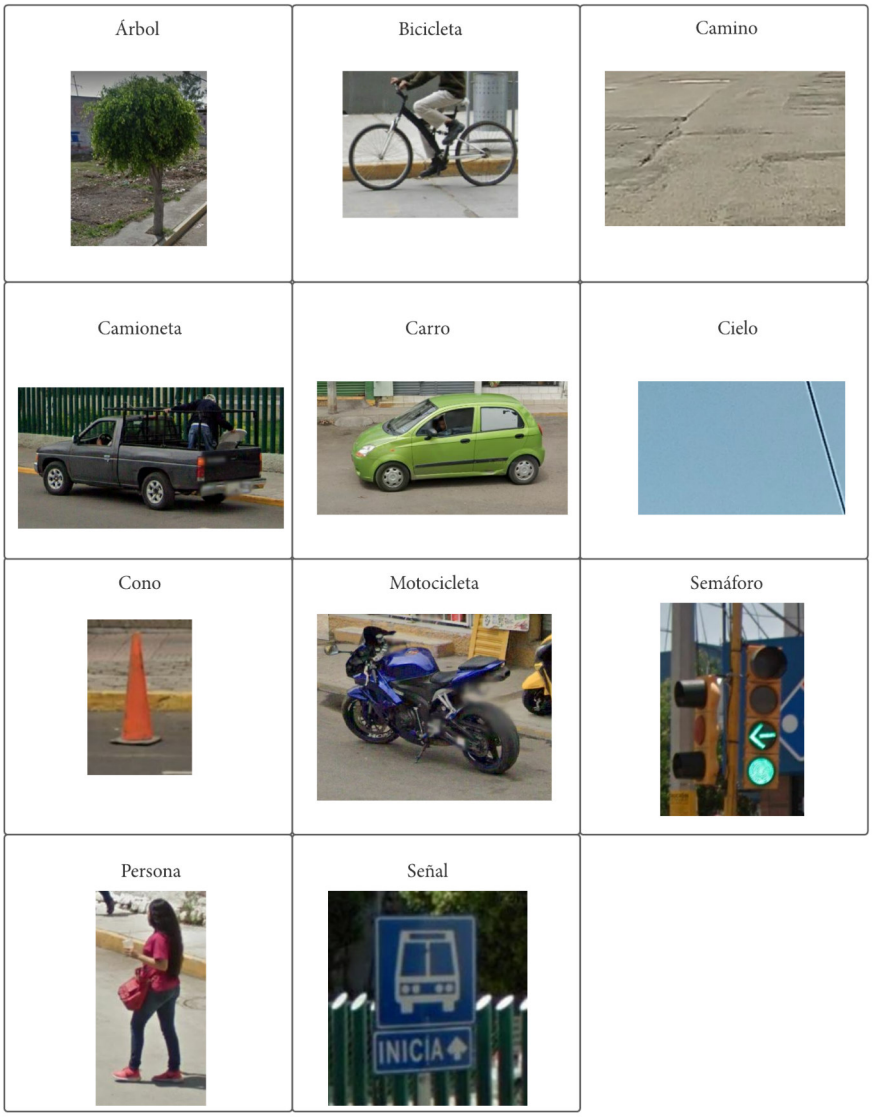

Figura 3.2: Adquisición de imágenes de cada clase.

En vista de que la mayoría de las imágenes se repetían, se buscó una ciudad dentro del estado de Guanajuato con un mayor flujo vial y con mayor variación de objetos en el escenario del vehículo. La segunda etapa de grabación se realizó en la ciudad de León, Gto., con una duración de una hora y media desde el sur de la ciudad hasta el norte de la ciudad, con un recorrido total de 13.4 km.

A su vez se grabaron 13 videos, y se aplicó la misma metodología de extraer una captura a 29 FPS, después se realizó el mismo proceso de máscara R-RNC, logrando obtener un conjunto de datos más completo y equilibrado como se muestra en la tabla 3.1.

Tabla 3.1: Total de muestras por clase.

Clases	No. de muestras
Árbol	2,681
Bicicleta	2,919
Camino	3,189
Camioneta	3,236
Carro	3,464
Cielo	3,062
Cono	3,001
Motocicleta	2,651
Persona	2,713
Semáforo	3,570
Señal	2,900
Total	33,387

3.3. Arquitecturas

En las siguientes secciones se hablará respecto a la construcción de cada una de las diferentes redes neuronales, describiendo el procedimiento que se llevó a cabo para el buen funcionamiento durante el entrenamiento y la validación del conjunto de datos de entrada.

3.3.1. AlexNet

Acerca de esta red se establecieron diferentes parámetros que marcan su arquitectura general y los cambios realizados a la red, como se muestra en la tabla 3.2. Comenzando con las dimensiones del conjunto de datos se

redimensionó el tamaño de las imágenes de entrada y quedaron de esta manera: 227 x 227 x 3. Estas imágenes entran a la primera capa mediante una convolución con función de activación ReLU y un filtro de 96, para después conectar con una normalización por lotes que estabiliza la salida de la convolución y mejora las características de entrada para el agrupamiento que procesará los datos con un tamaño de paso de 2. Cosa parecida se ejecuta en la segunda capa, iniciando al igual con una convolución mediante un filtro de 256 y un paso de 1 manteniendo la misma función de activación que la anterior capa, al igual se aplica una normalización por lotes y un agrupamiento para esta capa. En la tercera y cuarta capa sólo se implementó una convolución y una normalización por lotes respectivamente, con un filtro de 384 y un paso de 1. A la quinta capa se aplicó lo mismo que en la segunda capa, el filtro de convolución es de 256 y se mantuvo el mismo tamaño de paso, hasta este punto se tienen las características, ya que en la sexta capa y séptima capa se realiza el proceso para clasificación para las 11 clases y la función de activación softmax.

Tabla 3.2: Arquitectura propuesta AlexNet.

Capa	Tamaño del kernel	Pasos	Filtros
Entrada	227 x 247 x 3	—	—
Conv. (ReLU)	11	4	96
Normali. de lotes	—	—	—
Agrupamiento Máx.	3	2	—
Conv. (ReLU)	5	1	256
Agrupamiento Máx.	3	2	—
Normali. de lotes	—	—	—
Conv. (ReLU)	3	1	384
Normali. de lotes	—	—	—
Conv. (ReLU)	3	1	384
Normali. de lotes	—	—	—
Conv. (ReLU)	3	1	256
Normali. de lotes	—	—	—
Agrupamiento Máx.	3	2	—
Aplanado	—	—	—
Comple. Conect.	4096	—	—
Abandono	0.5	—	—
Comple. Conectada	4096	—	—
Abandono	0.5	—	—

Salida	11	—	softmax

3.3.2. ResNet 50 y 101

Acerca de esta red se establecieron diferentes parámetros que marcan su arquitectura general y los cambios realizados a la red, como se muestra en la tabla 3.3. Para la ResNet 50 y 101, en este caso se describirá la ResNet 50, ya que tiene mucho parecido con las ResNet 101 y en el punto en el que se note la diferencia se mencionará la diferencia para cada una. Comenzando con las dimensiones del conjunto de datos se redimensionó el tamaño de las imágenes de entrada y quedaron de esta manera 64 x 64 x 3 que entra a un relleno de ceros para el tamaño de la salida de las características, dentro de la primera capa se utiliza una convolución, conectada con una normalización por lotes para estabilizar la salida de la convolución con un tamaño de 3 y una función de activación ReLU, conectada con un agrupamiento máximo con una dimensión de 3 y un tamaño de paso de 2.

A partir de esta primera capa se realizan bloques de diferentes iteraciones y se denominarán como primer bloque y segundo bloque, continuando con el primer bloque que contiene tres capas más en el que la primera sub-capa tiene una convolución, una normalización por lotes. El segundo sub-bloque es similar al primero, la diferencia de sus parámetros con los que estará procesando las características y el tercer sub-bloque es una convolución y una normalización por lotes. Continuando con el segundo bloque que ejecuta la red, está formado con la misma estructura que el primer bloque, sólo que son parámetros constantes.

En ese mismo contexto se procede a incluir los parámetros necesarios para la segunda capa, la cual tiene el primer bloque y tres iteraciones del segundo bloque son 256 filtros, un tamaño de paso 1 y la dimensión de entrada del conjunto de datos es de 64, estos parámetros son iguales para las 2 redes. La tercera capa tiene cuatro iteraciones, está formada por el primer bloque

que le proceden y tres bloques del segundo con 512 filtros, un tamaño de paso de 2 y el tamaño del conjunto de imágenes de 128. En la cuarta capa es necesario resaltar la diferencia entre la ResNet 50 y la ResNet 101. Enseguida se explica para la ResNet 50, está formada por 6 iteraciones con 1024 filtros, un tamaño de paso de 2 y el tamaño del conjunto de datos es de 256. En segunda instancia para la ResNet 101 el número de iteraciones es de 23 y los parámetros son iguales a la ResNet 50. La quinta capa es igual en las dos redes y tienen una iteración de 3 con un tamaño de filtro de 2048, un tamaño de paso de 2 y la dimensión del conjunto de entrada es de 512.

A partir de este punto termina con la parte de extracción de características del modelo y la capa 6 y 7 se encarga de procesar las características para la clasificación de objetos, la capa 6 se desarrolla mediante un agrupamiento promedio con un tamaño de 2 y la capa 7 realiza un aplanamiento que conecta completamente para las 11 clases con una activación softmax.

Tabla 3.3: Arquitectura propuesta ResNet 50 y 101

1er bloque	Convolución, Normalización por lotes, Activación ReLu, Convolución, Normalización por lotes, Activación ReLu, Convolución, Normalización por lotes					
2do bloque	Convolución, Normalización por lotes, Activación ReLu, Convolución, Normalización por lotes, Activación ReLu, Convolución, Normalización por lotes, Iteración					
Capa	ResNet 50			ResNet 101		
	Filtros	Tam.	Conj. de datos	Filtro	Tam.	Conj. de datos
Entrada			64 x 64 x 3			64 x 64 x 3
Capa 1						
Conv.	7	1	64	7	1	64 x 64 x 3
N. de lotes	eje=3			eje =3		
Agrup.		3			3	
Capa 2						
1er bloque	3	256	64	3	256	64
2do bloque	3	256	64	3	256	64
Iteración 3						
Capa 3						
1er bloque	3	256	64	3	256	64
2do bloque	3	256	64	3	256	64
Iteración = 3						

Capa 4						
1er bloque	3	512	128	3	512	128
2do bloque	3	512	128	3	512	128
Iteración = 4						
Capa 5						
1er bloque	3	1024	256	3	1024	256
2do bloque	3	1024	256	3	1024	256
Iteración	6			23		
Capa 6						
1er bloque	3	2048	512	3	2048	512
2do bloque	3	2048	512	3	2048	512
Iteración = 3						
Capa 7						
Agrupamiento promedio =2						
Aplanamiento						
Completamente conectada, 11 clases, softmax						

3.3.3. VGGNET 16 y 19

Acerca de esta red se establecieron diferentes parámetros que marcan su arquitectura general y los cambios realizados a la red. Como se muestra en la tabla 3.4, comenzando con la dimensión del conjunto de datos que contiene un total, se redimensionó el tamaño de las imágenes de entrada y quedaron de esta manera, 224 x 224 x 3.

Se explicará de la misma forma que en la anterior red, cuando se llegue a la parte en la que hay diferencia entre la VGG16 y la VGG19 se hará mención de la diferencia entre las dos redes, ya que todos los demás parámetros son iguales en las ambas redes, además se menciona que cada capa tiene convolución y agrupamiento por lo cual se mencionará el número de convoluciones a partir de esto queda como establecido que el agrupamiento tiene un tamaño de 3 y un paso de 1.

La primera capa contiene 3 convoluciones con 64 filtros y un tamaño de kernel de 3 y el agrupamiento. La segunda capa tiene 2 convoluciones, 128

filtros y un tamaño de kernel de 3 y el agrupamiento. La tercera capa es igual a la segunda, lo único que cambia es el número de filtros de 128 a 256 y el agrupamiento es de 2. En la cuarta y quinta capa son iguales, pero para la VGG16 y VGG19 cambia, la VGG16 contiene 3 capas de convolución con 512 filtros y un tamaño de kernel de 3 que conecta con el agrupamiento y para la VGG19 tiene 4 convoluciones con los mismos parámetros que la de la VGG16 y al igual conecta con una capa de agrupamiento.

Por consiguiente, a partir de este punto termina con la parte de extraer las características del modelo, el cual contiene una etapa de aplanamiento que conecta con 2 capas completamente conectadas de 4096 y al final una activación softmax para las 11 clases.

Tabla 3.4: Arquitectura propuesta VGG 16 y 19.

Capa	VGG16			VGG19		
	Filtro	Tamaño K.	Paso	Filtros	Tamaño K.	Paso
Entrada 224 x 224 x 3						
Capa 1						
Conv. (3)	64	3	1	64	3	1
Agrup.		3	2		3	2
Capa 2						
Conv. (2)	128	3	1	128	3	1
Agrup.		3	2		3	2
Capa 3						
Conv. (2)	256	3	1	256	3	1
Agrup.		3	2		3	2
Capa 4						
Conv. (3) (4)	512	3	1	512	3	1
Agrup.		3	2		3	2
Capa 5						
Conv. (3) (4)	512	3	1	512	3	1
Agrup.		3	2		3	2
Capa 6						
Aplanamiento						
Completamente conectada = 4096						
Completamente conectada = 4096						
Completamente conectada, 11 clases, softmax						

3.3.4. LeNet5

Acerca de esta red se establecieron diferentes parámetros que marcan su arquitectura general y los cambios realizados a la red, como se muestra en la tabla 3.5, comenzando con las dimensiones del conjunto de datos que contiene un total, se redimensionó el tamaño de las imágenes de entrada y quedaron de esta manera, 32 x 32 x 3.

La primera capa tiene una convolución con 6 filtros, un tamaño de kernel de 5 y un tamaño de paso de 1 con una función de activación tangente hiperbólica que conecta a un agrupamiento promedio con un tamaño de 2 y un tamaño de paso de 2. La segunda capa tiene 16 filtros, un tamaño del kernel de 5 y un tamaño de paso de uno y al igual con la misma activación que la anterior, conectando con una capa de agrupamiento promedio con un tamaño de 2 y un tamaño de paso de 2. Por último, la tercera capa solo contiene una convolución con 120 filtros, un tamaño de kernel de 5 y un número de paso de 1. Con la misma función de activación que las anteriores hasta, este punto ya se tiene el procesamiento de las características y se procede con las capas para la clasificación, las cuales están conectadas a un aplanamiento y una función de activación softmax para las 11 clases.

Tabla 3.5: Arquitectura propuesta LeNet5.

Capa	Filtros	Tamaño del kernel	Pasos
Entrada = 32 x 32 x 3			
Conv. (Tanh)	6	5	1
Agrupamiento Máx.	2		2
Conv. (Tanh)	16	5	1
Agrupamiento Máx.	2		2
Conv. (Tanh)	120	5	1
Aplanamiento			
Completamente conectada, 11 clases, softmax			

Las diferentes arquitecturas explicadas en las secciones anteriores serán utilizadas en el siguiente capítulo y se explicarán los resultados obtenidos de cada arquitectura.

Capítulo 4

Resultados de las RNC

En el presente capítulo se describirán de manera detallada los resultados obtenidos durante la realización del presente proyecto, tales como la implementación para la detección de objetos basada en las arquitecturas descritas y explicadas en los dos capítulos anteriores.

4.1. Base de datos

En esta sección se describe el contenido de la base de datos y las técnicas que se usaron para la captura de datos, se determina como resultado, para este tipo de aplicación es mejor trabajar con datos de un ambiente real en el contexto de una ciudad a imágenes de un ambiente ideal, como lo fue el descargar imágenes de internet. La metodología de grabar y extraer los fotogramas para después procesarlos en una red neuronal convolucional basada en regiones, para obtener el objeto segmentando y extrayendo la región de interés para después colocar el objeto en un fondo negro y obteniendo una nueva imagen.

Se aumentó la base de datos con una amplia variedad de diferentes objetos que se tenían en cada clase, dando como resultado una base de datos con un total de 33,387 imágenes, la gran parte de las imágenes son de dimensiones medianas de los objetos, es necesario destacar que para esta metodología de grabar se necesitan ambientes concurridos y un flujo vial constante para lograr obtener una gran cantidad de objetos y con esta explicación de los resultados obtenidos mediante la base de datos se procede a presentar los resultados de las redes Neuronales convolucionales en las siguientes secciones.

En la sección de los resultados de cada RNC se utilizará el código de colores de la figura 4.1 para la localización de los objetos en las imágenes procesadas.

Figura 4.1: Etiquetado por colores para las clases seleccionadas.

4.2. AlexNet

En esta sección se describe al respecto de los resultado obtenidos en el entrenamiento y validación de la RNC AlexNet. Esta arquitectura se entrenó durante un total de 75 épocas, dando como resultados que pasando de la época 40 se acerca casi al 90 % como se muestra en la figura 4.2. Se puede decir que con un conjunto de datos más grande podría incrementar la exactitud de la validación.

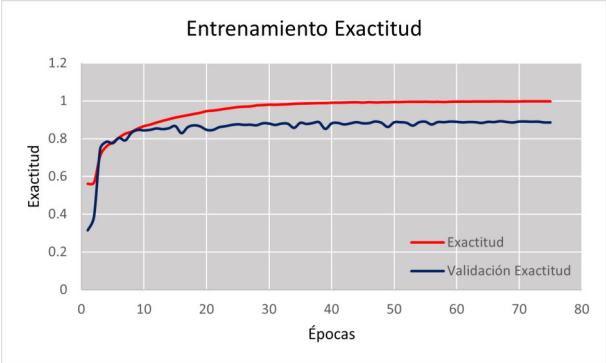

Figura 4.2: Gráfica de exactitud de la red AlexNet.

Respecto a las pérdidas como se muestra en la figura 4.3, basándose en el conjunto de datos que se ingresó a la RNC, se obtuvieron pérdidas entre 0.4 y 0.6 al igual que con la exactitud varía por la cantidad de imágenes ingresadas y podría acercarse más a las pérdidas de entrenamiento.

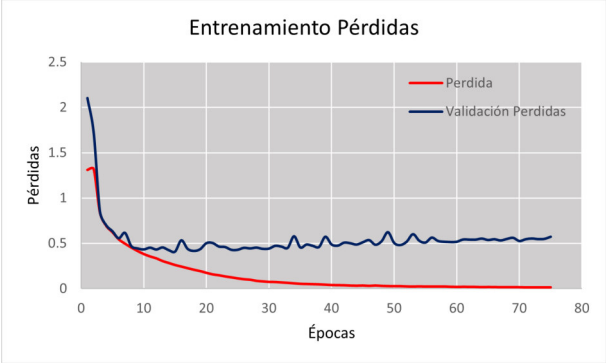

Figura 4.3: Gráfica de pérdidas de la red AlexNet.

De las gráficas anteriores también se calculó el resultado de la matriz de confusión como se muestra en la figura 4.4, arrojando los siguientes resultados que podrían tomarse como positivos en esta red.

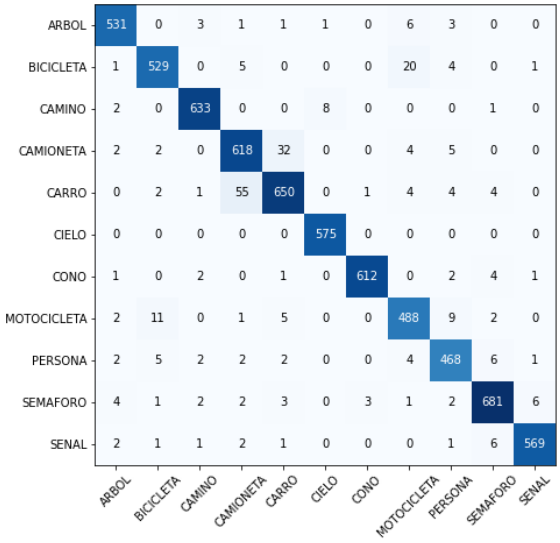

Figura 4.4: Matriz de confusión de la red AlexNet, los renglones es el esperado y las columnas es lo real.

Comenzando con los verdaderos positivos, ya que en la mayoría de las clases predice de forma correcta, tiene una confusión muy alta con las clases que son medios de transporte. En los falsos positivos la clase carro con camioneta tiene valores significativos que pueden afectar a la predicción, esto se da por el ambiente en el que se desarrolló el conjunto de datos y la forma de los objetos.

Dentro de los falsos negativos, la clase carro tiene la más alta confusión con 15 % de confusión con la clase camioneta, lo cual se puede entender como una muy alta similitud entre clases, al igual sucede lo mismo con la clase motocicleta con una confusión del 10 % con la clase bicicleta, esto se debe a que dentro de las imágenes del conjunto de datos el aprendizaje profundo encuentra similitud en el análisis repetitivo de las imágenes suponiendo que pertenece a una clase que no es correcta.

El resultado de la clasificación de objetos se demuestra que la RNC AlexNet

arroja buenos resultados de la predicción en la detección de objetos, como se muestra en las siguientes figuras 4.5, 4.6, 4.7 y 4.8.

Figura 4.5: Clasificación 1 AlexNet.

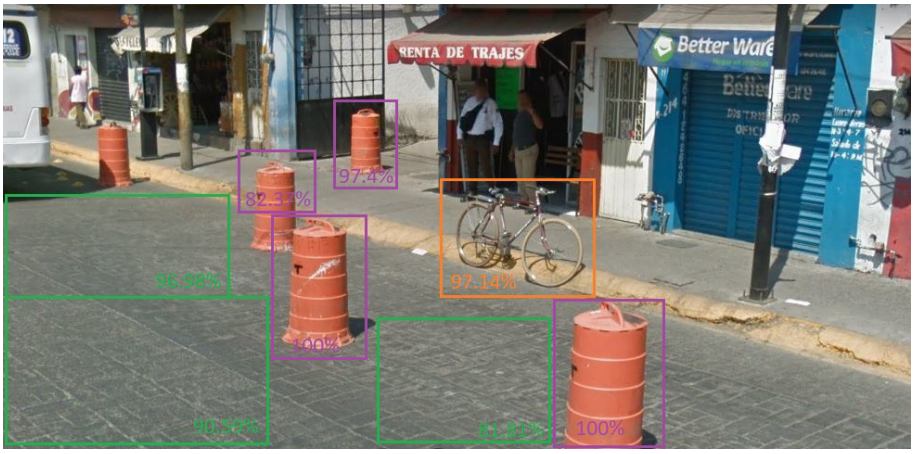

Figura 4.6: Clasificación 2 AlexNet.

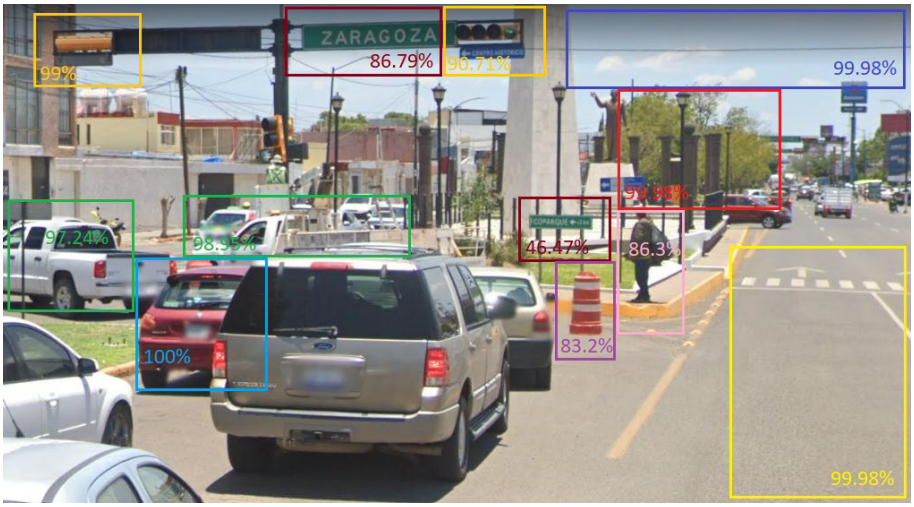

Figura 4.7: Clasificación 3 AlexNet.

Figura 4.8: Clasificación 4 AlexNet.

En los resultados se logra visualizar que las 11 clases son detectadas mediante un proceso de desfragmentar la imagen en 50 parches, dentro de cada figura se visualizan objetos que deberían ser reconocidos, y como se

explicó con la matriz de confusión hay objetos que confunde con otra clase muy similar al objeto como el ejemplo de carro - camioneta y persona - bicicleta.

Otro aspecto importante es la iluminación que tiene el objeto, provocando que no sea correctamente detectado, debido a la sombra que el objeto tiene.
Por último, la mayoría de los resultados están por encima del 80 % de exactitud, siendo la mejor red para la detección de las 11 clases.

4.3. LeNet5

En esta sección se hablará al respecto de los resultados obtenidos en el entrenamiento y validación de la RNC AlexNet durante un total de 900 épocas, dando como resultado que pasando de la época 500 se acerca casi al 75 % como se muestra en la figura 4.9, este modelo, no es de los más utilizados para la detección de objetos, pero si es usado en OCR.

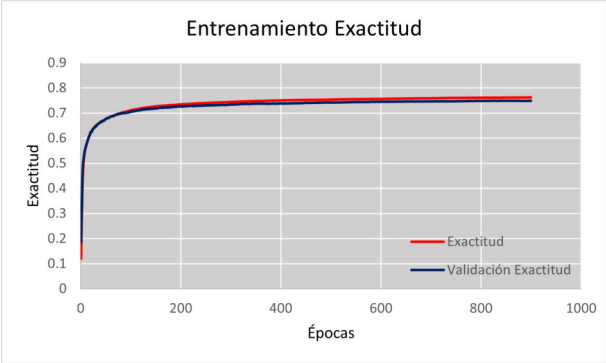

Figura 4.9: Gráfica de exactitud de la red LeNet5.

Respecto a las pérdidas, como se muestra en la figura 4.10, el conjunto

de datos que se ingresó a la RNC, se obtuvieron pérdidas entre 0.7 y 0.8 al igual que con la exactitud varía por la cantidad de imágenes ingresadas y podría acercarse más a las pérdidas de entrenamiento.

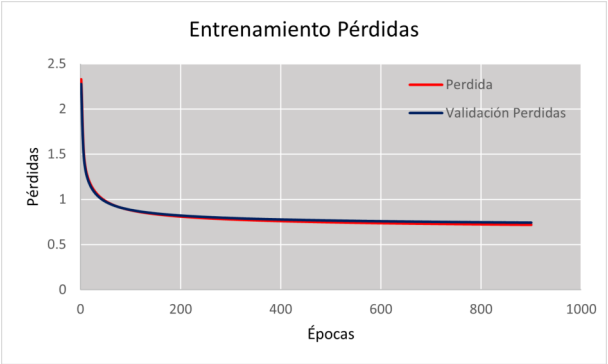

Figura 4.10: Gráfica de pérdidas de la red LeNet5.

De las gráficas anteriores también se calculó el resultado de la matriz de confusión como se muestra en la figura 4.11, arrojando los siguientes resultados que podrían tomarse como positivos en esta red.

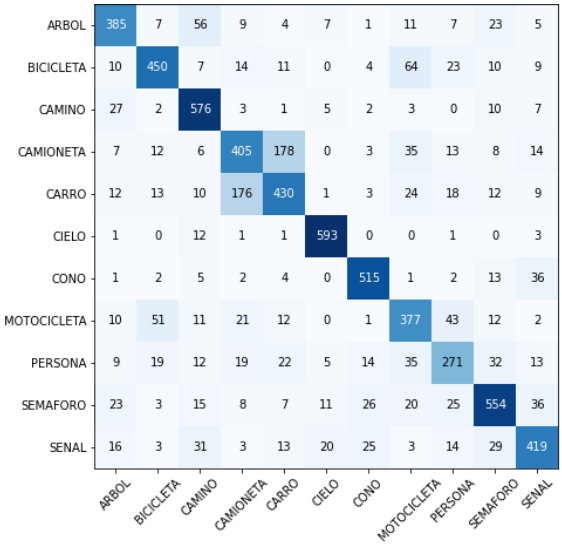

Figura 4.11: Matriz de confusión de la red LeNet5, los renglones es el esperado y las columnas es lo real.

Comenzando con los verdaderos positivos, ya que en la mayoría de las clases predice de forma correcta, tiene una confusión muy alta con la mayoría de las clases, esto se debe a que la red es el antepasado de las redes neuronales convolucionales y es usada para la detección de caracteres, por lo que la detección de objetos en ambientes urbanos confunde y realiza similitudes por colores y bordes.

Dentro de los falsos negativos, las clases carro y camioneta tienen la más alta confusión, con 25 % entre ambas clases, lo cual se puede entender como una muy alta similitud entre clases, al igual sucede lo mismo con las demás clases entre el 10 % y 15 % con las clases que tiene las mismas formas o colores, esto se debe a que las formas de los caracteres son continuas y varían poco.

El resultado de la clasificación de objetos se demuestra en la RNC LeNet5, arroja buenos resultados para la predicción en la detección de objetos, como se muestra en las siguientes figuras 4.12, 4.13, 4.14 y 4.15 para este análisis debemos recordar que esta red es más usada en aplicaciones de OCR.

En los resultados se logra visualizar que las 5 clases son detectadas mediante un proceso de desfragmentar la imagen en 50 parches, dejando 6 clases con resultados erróneos en la predicción de objetos, dentro de cada figura se visualizan objetos que deberían ser reconocidos, y como se explicó con la matriz de confusión, hay objetos que confunde con otra clase muy similar al objeto como carro - camioneta, persona - bicicleta y motocicleta - bicicleta.

Otro aspecto importante es la iluminación que tiene el objeto, generando que no sea correctamente detectado, debido a la sombra que el objeto tiene.

Figura 4.12: Clasificación 1 LeNet5.

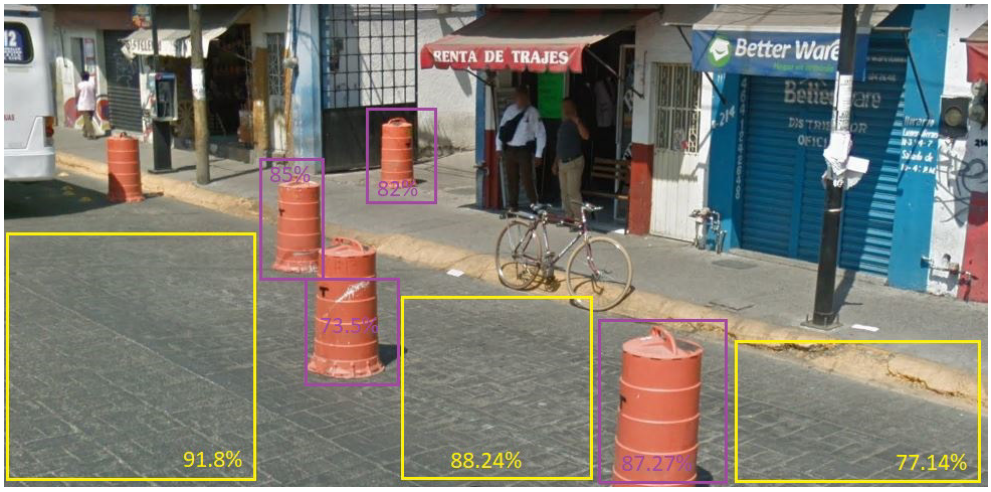

Figura 4.13: Clasificación 2 LeNet5.

Figura 4.14: Clasificación 3 LeNet5.

Figura 4.15: Clasificación 4 LeNet5.

Por último, las clases árbol, camino, cielo, y cono están por encima del 80 % de exactitud, la clase semáforo llega a tener variaciones menores al porcentaje de las anteriores clases, ocupando el último lugar en esta comparativa de redes neuronales convolucionales para la detección de las 11 clases.

4.4. ResNet 50

En esta sección se hablará al respecto de los resultados obtenidos en el entrenamiento y validación de la RNC ResNet 50 durante un total de 75 épocas, dando como resultado que pasando de la época 45 la exactitud deja de tener picos y se mantiene por encima del 80 % como se muestra en la figura 4.16, puedo decir que con un conjunto de datos más grande se podría incrementar la exactitud de la validación.

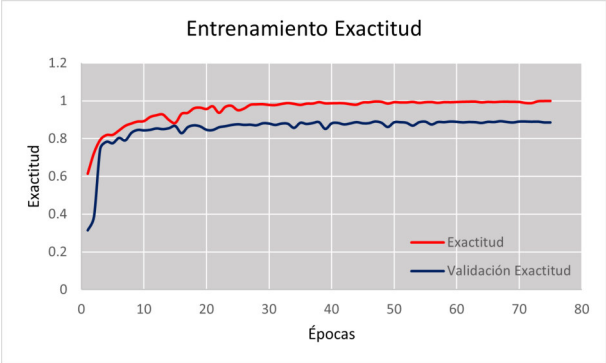

Figura 4.16: Gráfica de exactitud de la red ResNet 50.

Respecto a las pérdidas, como se muestra en la figura 4.17, el conjunto de datos que se ingresó a la RNC, se obtuvieron pérdidas de entre el 0.5 y 1.2 al igual que con la exactitud varía por la cantidad de imágenes ingresadas y podría acercarse más a las pérdidas de entrenamiento.

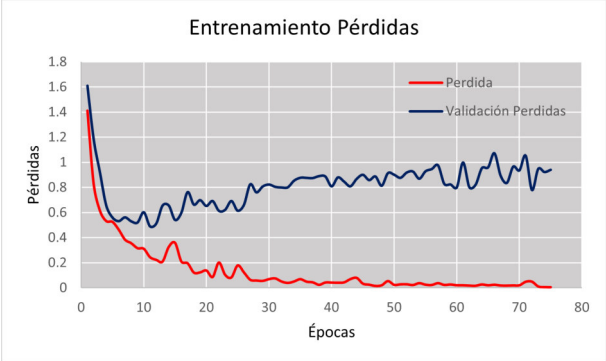

Figura 4.17: Gráfica de pérdidas de la red ResNet 50.

De las gráficas anteriores también se calculó el resultado de la matriz de confusión como se muestra en la figura 4.18, arrojando los siguientes resultados que podrían tomarse como positivos en esta red.

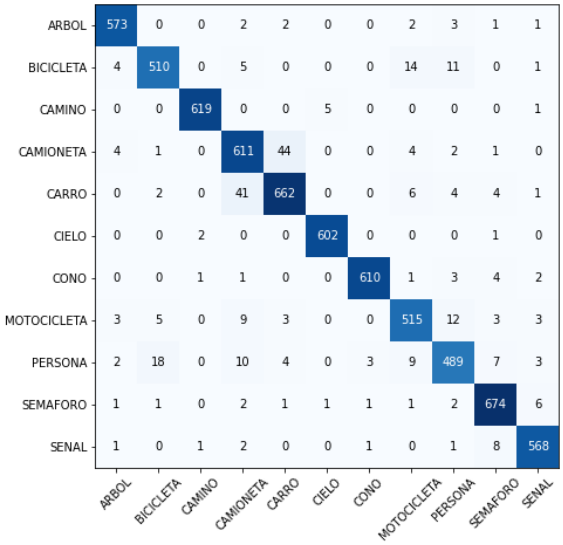

Figura 4.18: Matriz de confusión de la red ResNet 50, los renglones es el esperado y las columnas es lo real.

Comenzando con los verdaderos positivos y los falsos negativos, la mayoría de las clases son predichas de forma correcta, todavía tiene un cierto grado de confusión con algunas clases, como la clase carro, que tiene una confusión del 10 % con la clase camioneta. La clase bicicleta tiene una confusión del 8 % con la clase persona y la misma clase bicicleta tiene un 7 % con la clase motocicleta, la cual pueden afectar a la predicción, esto se da por el ambiente en el que se desarrolló el conjunto de datos y la forma del objeto.

El resultado de la clasificación de objetos se demuestra que la RNC ResNet 50 arroja buenos resultados de la predicción en la detección de objetos, como se muestra en las siguientes figuras 4.19, 4.20, 4.21 y 4.22.

Figura 4.19: Clasificación 1 ResNet 50.

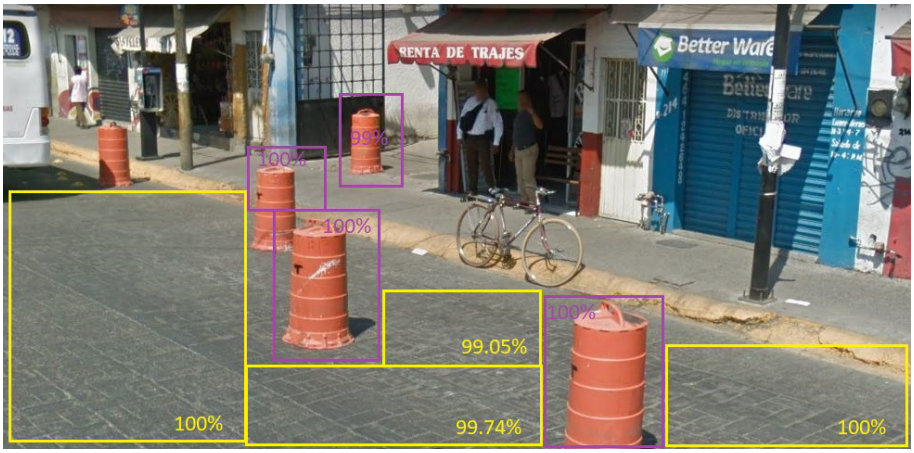

Figura 4.20: Clasificación 2 ResNet 50.

Figura 4.21: Clasificación 3 ResNet 50.

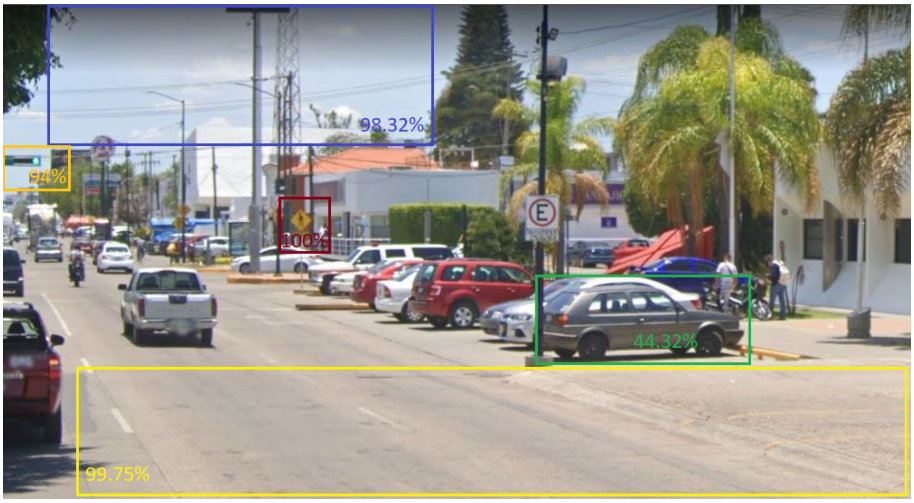

Figura 4.22: Clasificación 4 ResNet 50.

En los resultados se logra visualizar que las 6 clases son detectadas mediante un proceso de desfragmentar la imagen en 50 parches, dejando 5 clases con resultados erróneos en la predicción de objetos, dentro de cada figura

se visualizan objetos que deberían ser reconocidos, y como se explicó con la matriz de confusión hay objetos que confunde con otra clase muy similar al objeto como carro - camioneta, persona - bicicleta y motocicleta - bicicleta.

Otro aspecto importante es la iluminación que tiene el objeto, provocando que no sea correctamente detectado, debido a la sombra que el objeto tiene.
Por último, las clases árbol, camino, camioneta, cielo, cono y semáforo tienen una exactitud de 80 %, siendo la quinta mejor red para la detección de las 11 clases.

4.5. ResNet 101

En esta sección se hablará al respecto de los resultados obtenidos en el entrenamiento y validación de la RNC ResNet 101 durante un total de 45 épocas, dando como resultado que pasando de la época 25 la exactitud se equilibra y se mantiene en el 80 % como se muestra en la figura 4.23, puedo decir que con un conjunto de datos más grande podría incrementarse la exactitud de la validación.

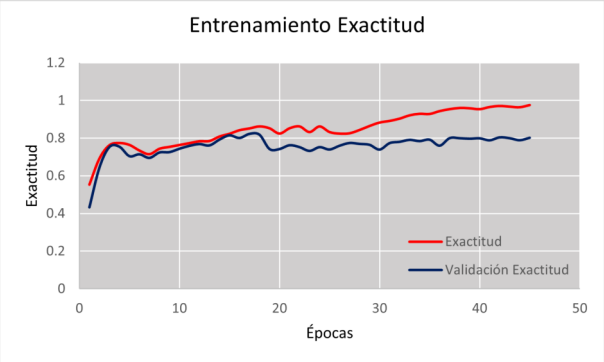

Figura 4.23: Gráfica de exactitud de la red ResNet 101.

Respecto a las pérdidas, como se muestra en la figura 4.24, el conjunto de datos que se ingresó a la RNC, se obtuvieron pérdidas entre 0.6 y 1.2, al igual que con la exactitud, varía por la cantidad de imágenes ingresadas y podría acercarse más a las pérdidas de entrenamiento.

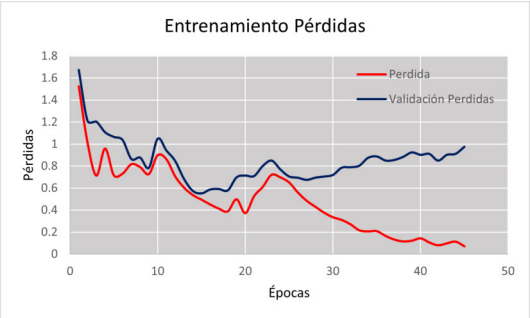

Figura 4.24: Gráfica de pérdidas de la red ResNet 101.

De las gráficas anteriores también se calculó el resultado de la matriz de confusión como se muestra en la figura 4.25, arrojando los siguientes resultados que podrían tomarse como positivos en esta red.

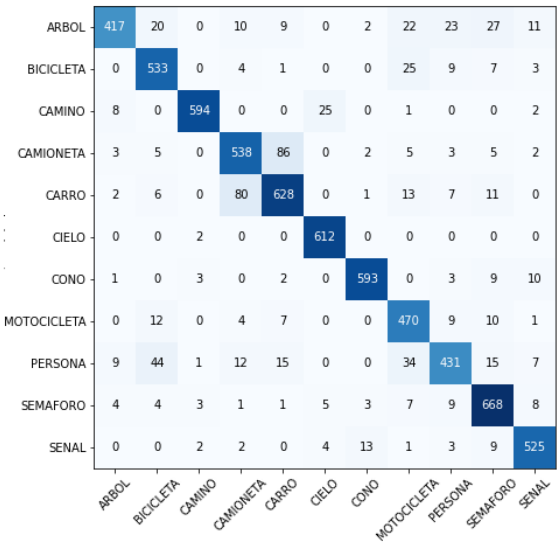

Figura 4.25: Matriz de confusión ResNet 101, los renglones es el esperado y las columnas es lo real.

Comenzando con los verdaderos positivos y los falsos negativos, la mayoría de las clases es predicha de forma correcta, todavía tiene un cierto grado de confusión con algunas clases, como la clase carro tiene una confusión del 15 % con la clase camioneta, la clase bicicleta tiene una confusión del 10 % con la clase persona y del 6 % con la clase motocicleta que pueden afectar a la predicción, esto se da por el ambiente en el que se desarrolló el conjunto de datos y la forma del objeto.

Del resultado de la clasificación de objetos se demuestra que la RNC ResNet 101 arroja mejores predicciones que ResNet 50 en la detección de objetos, y se muestran los resultados en las siguientes figuras 4.26, 4.27, 4.28 y 4.29. En los resultados se logra visualizar que las 7 clases son detectadas mediante un proceso de desfragmentar la imagen en 50 parches, dejando 4 clases con resultados erróneos en la predicción de objetos.

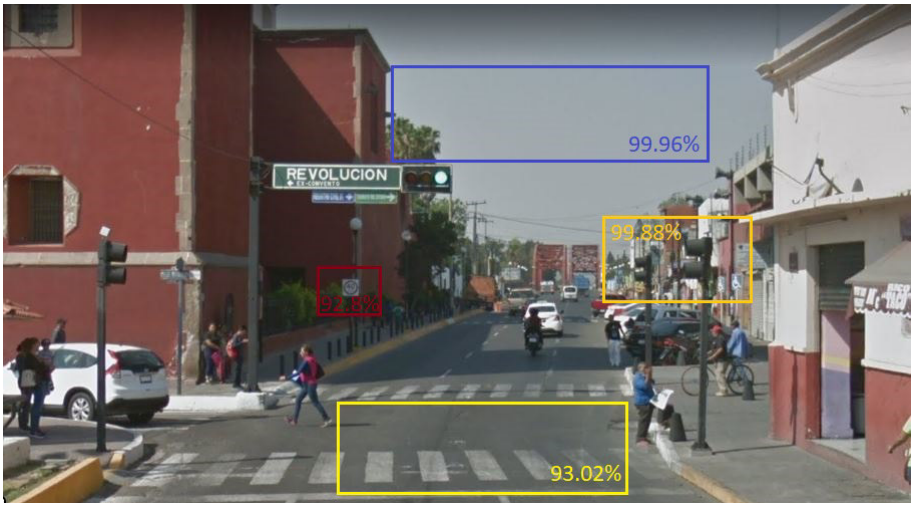

Figura 4.26: Clasificación 1 ResNet 101.

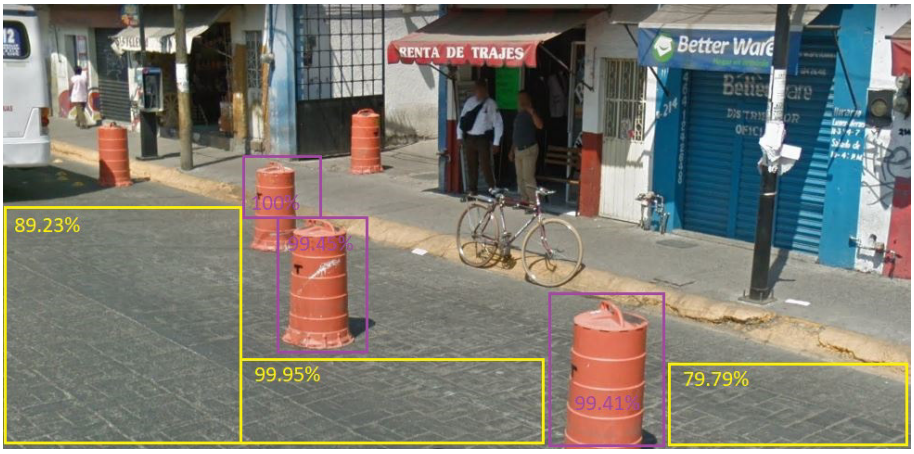

Figura 4.27: Clasificación 2 ResNet 101.

Figura 4.28: Clasificación 3 ResNet 101.

Figura 4.29: Clasificación 4 ResNet 101.

Otro aspecto importante es la iluminación que tiene el objeto, provocando que no sea correctamente detectado, debido a la sombra que el objeto tiene.

Por último, las clases árbol, camino, camioneta, cielo, cono y semáforo tienen una exactitud de 80 %, siendo la cuarta mejor red para la detección de las 11 clases.

4.6. VGG16

En esta sección se hablará al respecto de los resultados obtenidos en el entrenamiento y validación de la RNC VGG16 durante un total de 75 épocas, dando como resultado que pasando de la época 40 se acerca casi al 80 % como se muestra en la figura 4.30, puedo decir que con un conjunto de datos más grande podría incrementarse la exactitud de la validación.

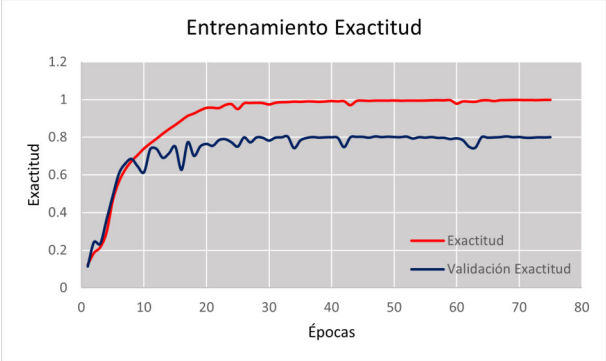

Figura 4.30: Gráfica de exactitud de la red VGG16.

Respecto a las pérdidas, como se muestra en la figura 4.31, el conjunto de datos que se ingresó a la RNC, se obtuvieron pérdidas entre 0.8 y 1.7 al igual que con la exactitud varía por la cantidad de imágenes ingresadas y podría acercarse más a las pérdidas de entrenamiento.

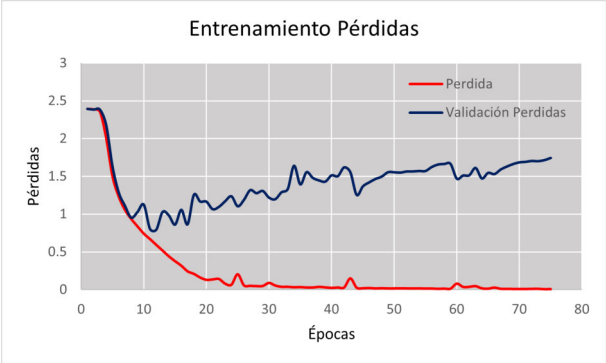

Figura 4.31: Gráfica de pérdidas de la red VGG16.

De las gráficas anteriores también se calculó el resultado de la matriz de confusión como se muestra en la figura 4.32, arrojando los siguientes resultados que podrían tomarse como positivos en esta red.

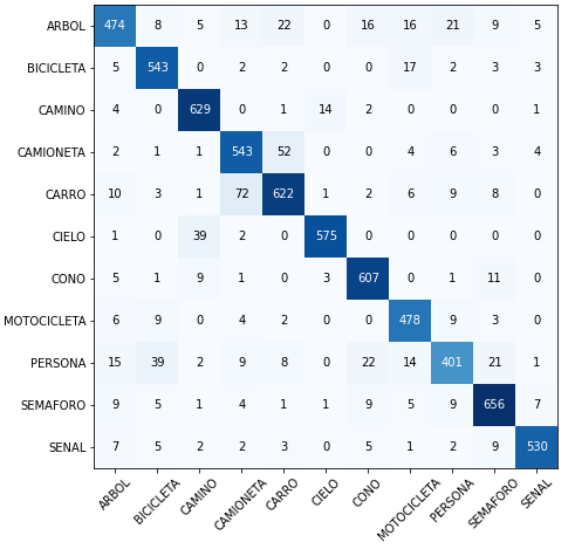

Figura 4.32: Matriz de confusión VGG16, los renglones es el esperado y las columnas es lo real.

Comenzando con los verdaderos positivos, ya que en la mayoría de las clases predice de forma correcta, todavía tiene un cierto grado de confusión con algunas clases, como podría ser las clases carro y camioneta. En los falsos positivos las clases, camioneta y carro tienen el 12 % de confusión, esto se da por el ambiente en el que se desarrolló el conjunto de datos y la forma del objeto.

Dentro de los falsos negativos, las clases bicicleta y persona tienen un 12 % de confusión al predecir el objeto, esto se debe a que dentro de las imágenes del conjunto de datos el aprendizaje profundo encuentra similitud en el análisis repetitivo de las imágenes, suponiendo que pertenece a una clase que no es correcta.

Del resultado de la clasificación de objetos se demuestra que la RNC VGG16 arroja mejores predicciones que ResNet 101 en la detección de objetos, y se muestran los resultados en las siguientes figuras 4.33, 4.34, 4.35 y 4.36. En los resultados se logra visualizar que las 7 clases son detectadas mediante un proceso de desfragmentar la imagen en 50 parches, dejando 4 clases con resultados erróneos en la predicción de objetos, dentro de cada figura se visualizan objetos que deberían ser reconocidos, y como se explicó con la matriz de confusión hay objetos que confunde con otra clase muy similar al objeto como carro - camioneta, persona - bicicleta y motocicleta - bicicleta.

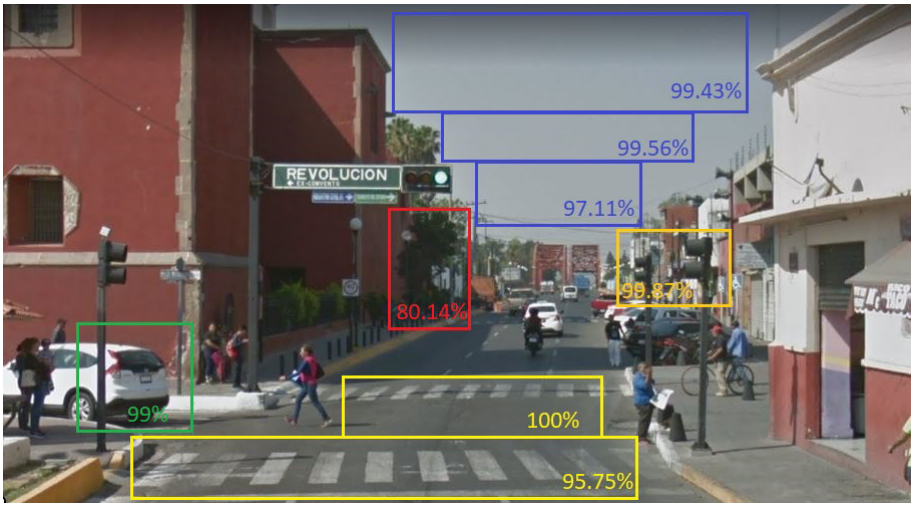

Figura 4.33: Clasificación 1 VGG16.

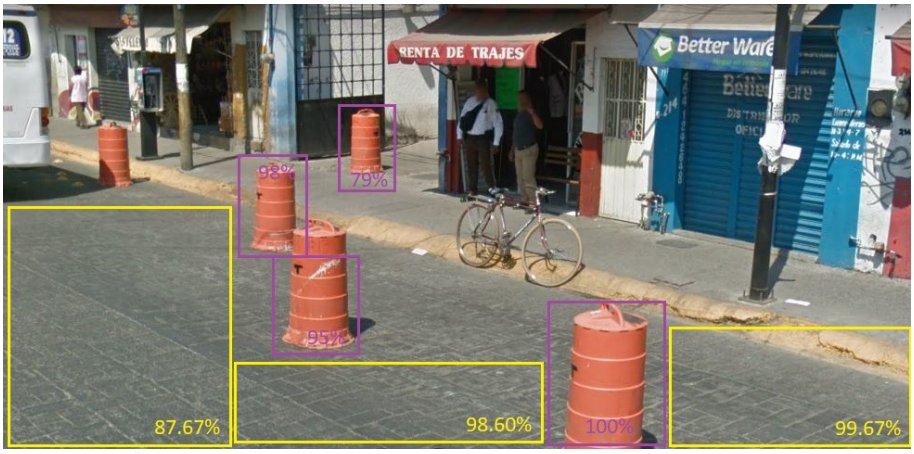

Figura 4.34: Clasificación 2 VGG16.

Figura 4.35: Clasificación 3 VGG16.

Figura 4.36: Clasificación 4 VGG16.

Otro aspecto importante es la iluminación que tiene el objeto, provocando que no sea correctamente detectado, debido a la sombra que el objeto tiene.

Por último, las clases árbol, camino, camioneta, cielo, cono, persona y semáforo tienen una exactitud de 80 %, siendo la tercera mejor red para la detección de las 11 clases.

4.7. VGG19

En esta sección se hablará al respecto de los resultados obtenidos en el entrenamiento y validación de la RNC VGG19 durante un total de 45 épocas, dando como resultado que pasando de la época 30 se acerca al 80 % como se muestra en la figura 4.37, puedo decir que con un conjunto de datos más grande podría incrementarse la exactitud de la validación.

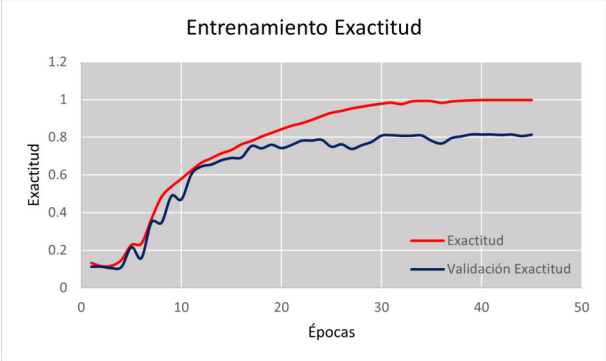

Figura 4.37: Gráfica de exactitud de la red VGG19.

Respecto a las pérdidas, como se muestra en la figura 4.38, el conjunto de datos que se ingresó a la RNC, se obtuvieron pérdidas entre 0.7 y 1.4 al igual que con la exactitud varía por la cantidad de imágenes ingresadas y podría acercarse más a las pérdidas de entrenamiento.

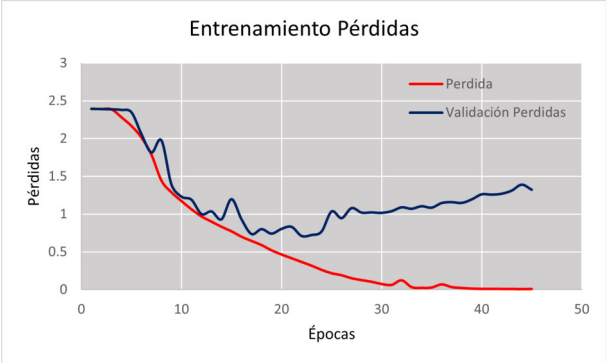

Figura 4.38: Gráfica de pérdidas de la red VGG19.

De las gráficas anteriores también se calculó el resultado de la matriz de confusión como se muestra en la figura 4.39, arrojando los siguientes resultados que podrían tomarse como positivos en esta red. Comenzando con

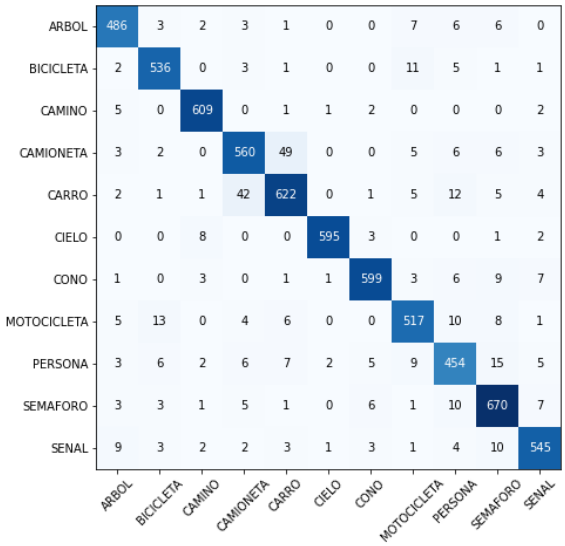

Figura 4.39: Matriz de confusión VGG19, los renglones es el esperado y las columnas es lo real.

los verdaderos positivos, ya que en la mayoría de las clases son predichas de forma correcta, todavía tiene un cierto grado de confusión con algunas clases, como podría ser las clases carro y camioneta. En los falsos positi-

vos la clase carro y camioneta tienen el 11 %, de confusión, esto se da por el ambiente en el que se desarrolló el conjunto de datos y la forma del objeto.

Dentro de los falsos negativos, las clases motocicleta y bicicleta tienen un 6 % de confusión al predecir el objeto, esto se debe a que dentro de las imágenes del conjunto de datos el aprendizaje profundo encuentra similitud en el análisis repetitivo de las imágenes, suponiendo que pertenece a una clase que no es correcta.

El resultado de la clasificación de objetos se demuestra que la RNC VGG19 arroja mejores predicciones que VGG16 en la detección de objetos, y se muestran los resultados en las siguientes figuras 4.40, 4.41, 4.42 y 4.43.

Figura 4.40: Clasificación 1 VGG19.

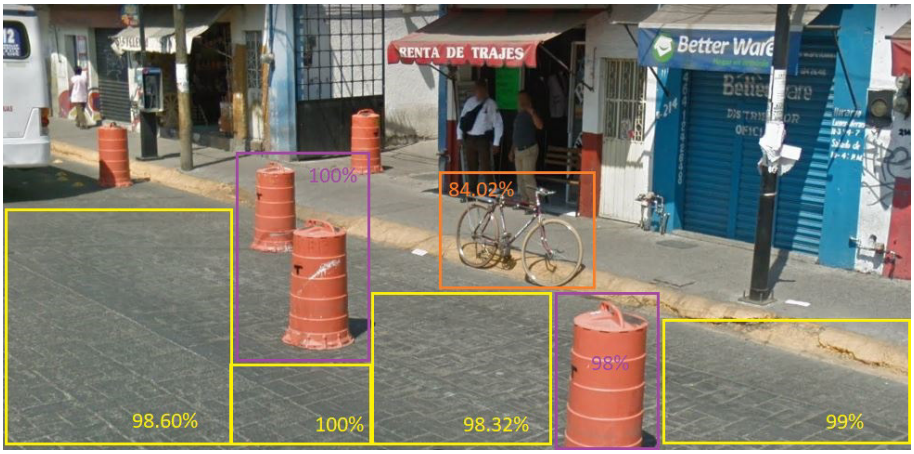

Figura 4.41: Clasificación 2 VGG19.

Figura 4.42: Clasificación 3 VGG19.

Figura 4.43: Clasificación 4 VGG19.

En los resultados se logra visualizar que las 11 clases son detectadas mediante un proceso de desfragmentar la imagen en 50 parches, dentro de cada figura se visualizan objetos que deberían ser reconocidos, y como se explicó con la matriz de confusión hay objetos que confunde con otra clase muy similar al objeto como carro - camioneta, persona - bicicleta y motocicleta - bicicleta.

Otro aspecto importante es la iluminación que tiene el objeto, provocando que no sea correctamente detectado, debido a la sombra que el objeto tiene.
Por último, las clases árbol, camino, camioneta, cielo, cono, persona y semáforo tienen una exactitud de 80 %, siendo la segunda mejor red analizada para la detección de las 11 clases.

En la tabla 4.1 se mostrarán los resultados resumidos de la predicción esperada mediante porcentaje que tuvo cada una de las RNC con las 11 clases a clasificar.

Tabla 4.1: Resultados de las predicciones de cada RNC

Clase	AlexNet	LeNet5	ResNet50	ResNet101	VGG16	VGG19
Árbol	97.25 %	74.76 %	98.12 %	77.08 %	80.61 %	94.55 %
Bicicleta	94.46 %	74.75 %	93.58 %	91.58 %	94.11 %	95.71 %
Camino	98.29 %	90.57 %	99.04 %	94.29 %	96.62 %	98.23 %
Camioneta	93.21 %	59.47 %	91.60 %	82.90 %	88.15 %	88.33 %
Carro	90.15 %	60.73 %	91.94 %	83.96 %	84.74 %	89.50 %
Cielo	100.00 %	96.90 %	99.50 %	99.67 %	93.19 %	97.70 %
Cono	98.23 %	88.64 %	98.07 %	95.49 %	95.14 %	95.08 %
Motocicleta	94.21 %	69.81 %	93.13 %	91.62 %	93.54 %	91.67 %
Persona	95.12 %	60.09 %	89.72 %	75.88 %	75.38 %	88.33 %
Semáforo	96.60 %	76.10 %	97.68 %	93.69 %	92.79 %	94.77 %
Señal	97.60 %	72.74 %	97.59 %	93.92 %	93.64 %	93.48 %

Capítulo 5

Conclusiones

El AP se está volviendo parte de nuestra vida cotidiana, cada vez la IA nos rodea y a la vez nos beneficia para facilitar las actividades que realizamos, el claro ejemplo lo tenemos con la conducción autónoma, la cual es una aplicación de la IA.

La conducción autónoma ha disminuido los accidentes viales, ha aumentado el tiempo de productividad de las personas que usan estos vehículos. Las tecnologías que se aplican en estos vehículos también han avanzado junto con esta aplicación, al igual que las RNC. Con esto puedo concluir el incremento exponencial que este campo ha tenido mediante las diferentes técnicas y métodos que se aplican, logran mejorar los resultados para la detección, localización y clasificación de los objetos que captan mediante los sensores de visión.

También las diferentes formas de procesar las imágenes del sensor han avanzado, como se sabe hay RNC que se dedican a una sola aplicación u otras redes que realizan varios procesos en una sola red, podría dejar el tema abierto en el que el día de mañana los vehículos de conducción autónoma dentro de los ambientes por los que pasen funcionen sólo con sensores de

visión, dejando atrás los distintos sensores que permiten la trayectoria en los ambientes por los que transcurre.

Por consiguiente, la construcción de un conjunto de datos puedo decir que es el punto más importante para obtener buenos resultados en una RNC, ya que si el conjunto, tiene defectos, o fallas en las imágenes a procesar los resultados serán incorrectos, de este punto se pueden realizar diferentes procesos para la adquisición del conjunto.

El aplicar las diferentes redes de aprendizaje profundo para la detección de objetos me abre la mente a pensar en diferentes aplicaciones en las que se podría utilizar, al igual para las nuevas redes que vienen por delante y seguirán evolucionando su arquitectura y el procesamiento de las características.

Por último, se reconoce el buen funcionamiento de las RNC, ya que de las diferentes arquitecturas que existen se destaca las redes AlexNet y VGG por su funcionalidad en la detección de objetos, aunque en las 11 clases que se tenían fueron las que mejor procesaron y clasificaron de forma correcta la mayoría de los objetos.

La mayoría de las clases tenían confusión con las clases carro y camioneta por la similitud de características que tienen las dos clases o la clase persona y bicicleta por el hecho de tener a una persona montado en una bicicleta, y la última confusión que se desarrolló fue con la clase bicicleta y motocicleta al igual que carro y camioneta fue por la similitud de las formas y figuras que tienen las motocicletas y bicicletas.

Bibliografía

[1] T. Inagaki and T. Sheridan, "A critique of the sae conditional driving automation definition, and analyses of options for improvement," *Cognition, Technology and Work*, pp. 1–5, 2018. 10.1007/s10111-018-0471-5.

[2] H. Tech, "Decoding the dichotomy: Traditional image processing vs. deep learning.," *Traditional Image Processing vs. Deep Learning*, pp. 1–6, 2020.

[3] M. Al-Qizwini, I. Barjasteh, H. Al-Qassab, and H. Radha, "Deep learning algorithm for autonomous driving using googlenet," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 89–96, IEEE, 2017. 10.1109/IVS.2017.7995703.

[4] L. Wang, X. Fan, J. Chen, J. Cheng, J. Tan, and X. Ma, "3d object detection based on sparse convolution neural network and feature fusion for autonomous driving in smart cities," *Sustainable Cities and Society*, vol. 54, p. 102002, 2020. 10.1016/j.scs.2019.102002.

[5] J. Zhang, Q. Su, C. Wang, and H. Gu, "Monocular 3d vehicle detection with multi-instance depth and geometry reasoning for autonomous driving," *Neurocomputing*, vol. 403, pp. 182–192, 2020. 10.1016/j.neucom.2020.03.076.

[6] D.-S. Hong, H.-H. Chen, P.-Y. Hsiao, L.-C. Fu, and S.-M. Siao, "Crossfusion net: Deep 3d object detection based on rgb images and point

clouds in autonomous driving," *Image and Vision Computing*, vol. 100, p. 103955, 2020. 10.1016/j.imavis.2020.103955.

[7] D. Chaves, S. Saikia, L. Fernández-Robles, E. Alegre, and M. Trujillo, "Una revisión sistemática de métodos para localizar automáticamente objetos en imágenes," *Revista Iberoamericana de Automática e Informática industrial*, vol. 15, no. 3, pp. 231–242, 2018. 10.4995/riai.2018.10229.

[8] S. A. Bagloee, M. Tavana, M. Asadi, and T. Oliver, "Autonomous vehicles: challenges, opportunities, and future implications for transportation policies," *Journal of modern transportation*, vol. 24, no. 4, pp. 284–303, 2016. 10.1007/s40534-016-0117-3.

[9] A. Rosebrock, *Deep Learning for Computer Vision*, ch. How Deep Is Deep? PYIMAGE SEARCH, 1 ed., 2018.

[10] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*, pp. 1–6, Ieee, 2017. 10.1109/ICEngTechnol.2017.8308186.

[11] A. Rosebrock, *Deep Learning for Computer Vision*, ch. Convolutional Neural Networks. PYIMAGE SEARCH, 1 ed., 2018.

[12] Krizhevsky, A. Nair, V. Hinton, G. , "Cifar-10 and cifar-100 (canadian institute for advanced research)." http://www.cs.toronto.edu/ kriz/cifar.html.

[13] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[14] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *towards data science*, vol. 6, no. 12, pp. 310–316, 2017. 10.33564/ijeast.2020.v04i12.054.

[15] J. Brownlee, *Better deep learning: train faster, reduce overfitting, and make better predictions*. Machine Learning Mastery, 2018.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015. 10.1109/CVPR.2015.7298594.

[18] W. Yu, K. Yang, Y. Bai, T. Xiao, H. Yao, and Y. Rui, "Visualizing and comparing alexnet and vgg using deconvolutional layers," in *Proceedings of the 33 rd International Conference on Machine Learning*, 2016.

[19] J. Wei, "Alexnet: The architecture that challenged cnns," 2019. https://towardsdatascience.com/alexnet-the-architecture-that-challenged-cnns-e406d5297951.

[20] R. A. Minhas, A. Javed, A. Irtaza, M. T. Mahmood, and Y. B. Joo, "Shot classification of field sports videos using alexnet convolutional neural network," *Applied Sciences*, vol. 9, no. 3, p. 483, 2019. 10.3390/app9030483.

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

[22] S. Targ, D. Almeida, and K. Lyman, "Resnet in resnet: Generalizing residual architectures," *arXiv preprint arXiv:1603.08029*, 2016. 10.21437/Interspeech.2020-2039.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 10.1109/CVPR.2016.90.

[24] A. Bouti, M. A. Mahraz, J. Riffi, and H. Tairi, "A robust system for road sign detection and classification using lenet architecture based on convolutional neural network," *Soft Computing*, vol. 24, no. 9, pp. 6721–6733, 2020. 10.1007/s00500-019-04307-6.

[25] S. Saxena, "The architecture of lenet-5," March 2021. https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/.

[26] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014. 10.1109/CVPR.2014.81.

[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015. 10.1109/TPAMI.2016.2577031.

[28] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015. 10.1109/ICCV.2015.169.

[29] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017. 10.1109/TPAMI.2018.2844175.

[30] W. Tahir, A. Majeed, and T. Rehman, "Indoor/outdoor image classification using gist image features and neural network classifiers," in *2015 12th International Conference on High-Capacity Optical Networks and Enabling/Emerging Technologies (HONET)*, pp. 1–5, IEEE, 2015. 10.1109/HONET.2015.7395428.

[31] M. Szummer and R. W. Picard, "Indoor-outdoor image classification," in *Proceedings 1998 IEEE International Workshop on Content-Based Access of Image and Video Database*, pp. 42–51, IEEE, 1998. 10.1109/CAIVD.1998.646032.

[32] J. Luo and A. Savakis, "Indoor vs outdoor classification of consumer photographs using low-level and semantic features," in *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, vol. 2, pp. 745–748, IEEE, 2001.

[33] L. Zhang, M. Li, and H.-J. Zhang, "Boosting image orientation detection with indoor vs. outdoor classification," in *Sixth IEEE Workshop on Applications of Computer Vision, 2002.(WACV 2002). Proceedings.*, pp. 95–99, IEEE, 2002. 10.1109/ACV.2002.1182164.

[34] P. Fitzpatrick, "Indoor/outdoor scene classification project," *Pattern Recognition and Analysis*, 2015.

[35] A. Olafenwa, "Semantic segmentation of videos with pixellib using ade20k model," 2020. https://pixellib.readthedocs.io/en/latest/index.html.

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014. 10.1007/978-3-319-10602-1 48.