

Detección de la señalización de tránsito vertical con redes neuronales convolucionales basadas en bloques residuales

Detection of vertical traffic signaling with convolutional neural networks based on residual blocks

Adrian Javier Alarcon Vargas¹

**Instituto de Investigaciones en Ciencia y Tecnología,
Universidad La Salle, Bolivia**
ajalarconvargas@gmail.com

Artículo Recibido: 23-04-22

Artículo Aceptado: 25-07-22

Resumen

El objetivo del presente trabajo es entrenar una red neuronal capaz de detectar la señalización de tránsito vertical y clasificarla usando bloques residuales. La metodología utilizada para el desarrollo de la red neuronal comprende cuatro fases: definición de la red neuronal, entrenamiento, utilización y mantenimiento de la red neuronal. Para el desarrollo de la red neuronal se cuenta con dos *datasets*, el primero es de origen alemán, consta de 50.000 imágenes y es muy usado para la clasificación de señales de tránsito; y el segundo de origen boliviano, que tiene 9.548 imágenes de carretera. El porcentaje de eficacia de la red neuronal nro. 1 con el *dataset* *GTSRB* es alto, obteniendo un valor de 94.36%, además incluye valores altos en el reporte de clasificación, caso contrario sucede con el *dataset* de Bolivia debido a que el *dataset* está desbalanceado.

Palabra Clave:

GTSRB, Segmentación, Procesamiento de Imágenes, Accidentes de Tránsito, Clasificación, *ResNet*, *ResUnet*.

¹ Ingeniero de Sistemas con especialización en desarrollo de aplicaciones web y aplicaciones móviles. Miembro del Instituto de Investigación en Ciencia y Tecnología de la Universidad La Salle. ORCID: <https://orcid.org/0000-0002-4716-4112>

Abstract

The objective of the present work is to train a neural network capable of detecting vertical traffic signaling and classify it using residual blocks, as they allow deeper neural networks. The methodology used for the development of the neural network comprises four phases: neural network definition, training, utilization, and maintenance of the neural network. For the development of the neural network there are two datasets, the first is of German origin, consists of 50,000 images and is widely used for the classification of traffic signs; and the second of Bolivian origin, which has 9,548 road images. The percentage of efficiency of the neural network no. 1 with the GTSRB dataset is high, obtaining a value of 94.36%, it also includes high values in the classification report, otherwise, it happens with the Bolivia dataset because the dataset is unbalanced.

Keywords:

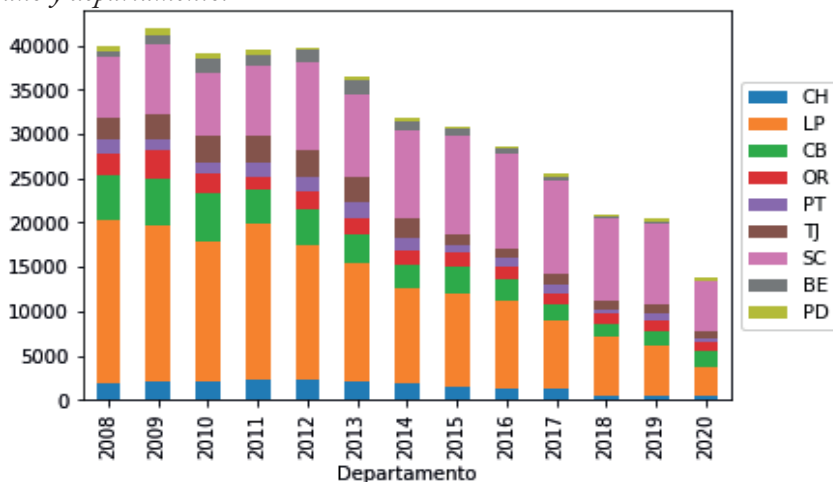
GTSRB, Segmentation, Image Processing, Traffic Accidents, Classification, ResNet, ResUnet.

Introducción

Un problema que es muy recurrente en Bolivia y en el mundo son los accidentes de tránsito, si bien se ha visto ciertas medidas para reducirlas, como por ejemplo la refacción de carreteras, creación de carreteras doble vía o el incremento de oficiales de tránsito, aun se puede evidenciar un alto número de accidentes. Según el Instituto Nacional de Estadística (2020), entre los años 2008 y 2020 se registró un total de 408.493 accidentes de tránsito en Bolivia, generando un total de 17.148 personas fallecidas. Si bien se tuvo una disminución de estos eventos causada por la pandemia, como se muestra en la Figura 1, con la vuelta a la normalidad los accidentes de tráfico podrían aumentar, causando inseguridad en la población. Hoy en día se tiene grandes avances en el uso de redes neuronales para el procesamiento de imágenes, por lo cual es necesario implementar soluciones a este problema.

Figura 1

Gráfico de barras apiladas basado en la cantidad de accidentes de tránsito por año y departamento.



Fuente: Elaboración Propia basado en (Instituto Nacional de Estadística, 2020)

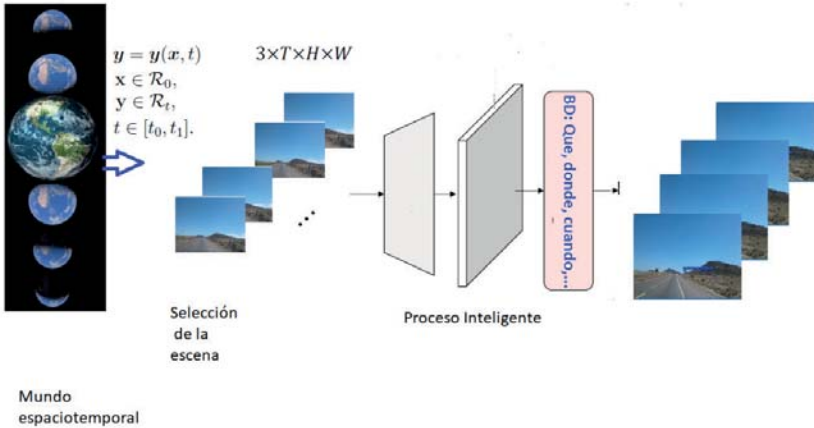
El presente trabajo tiene como fin entrenar una red neuronal que sea capaz de detectar la señalización de tránsito vertical y clasificarla con el fin de reducir accidentes de tránsito y reducir el temor de la población sobre este tema. Esta herramienta podría tener una infinidad de usos como por ejemplo avisar a un conductor cuando está por cometer algún acto por la cual recibiría una infracción o causaría un accidente de tránsito, una implementación en alguna empresa de industria automotriz, entre otras con el fin de reducir los accidentes de tránsito y reducir la inseguridad de la población en este ámbito.

Para el desarrollo de la red neuronal se tiene como entrada una selección de escenas dadas gracias al mundo espaciotemporal, que es el modelo que combina el espacio y el tiempo, en él se representan todos los acontecimientos físicos. A través de estas escenas y mediante un proceso inteligente se puede obtener como resultado la reconstrucción de la misma

escena a través de videos con una segmentación y clasificación de las señales de tránsito encontradas. El modelo de este proceso se ve en la figura 2.

Figura 2

Reconstrucción de un escenario de cruce de vía en formato de video.



Fuente: (Reynaga y Yupanqui, 2022)

Este trabajo se basa en dos *datasets* uno de origen boliviano y otro del origen alemán, también se usó las arquitecturas de red neuronal *ResNet* para la clasificación de señales y la arquitectura *ResUnet* para la segmentación.

Señalización Vertical en Bolivia

La señalización consiste en un conjunto de símbolos, marcas y figuras que tienen por objeto transmitir el mensaje oportuno a los usuarios para prevenir de hechos de tránsito e infracciones (Automóvil Club Boliviano, 2004). Las señales verticales son placas fijadas en postes, que, mediante símbolos, cumplen la función de prevenir sobre peligros, reglamentar las prohibiciones y brindar la información necesaria para guiar a los usuarios. (APIA XXI Ingenieros y Arquitectos Consultores, 2004).

Señalización vertical preventiva. Para APIA XXI Ingenieros y Arquitectos Consultores (2004), tienen como propósito advertir a los usuarios de la existencia de riesgos y/o situaciones especiales en la vía o en sus zonas adyacentes. En general, tienen la forma de un cuadrado con una de sus diagonales colocada verticalmente. Su color de fondo es amarillo.

Señalización vertical reglamentaria. En (APIA XXI Ingenieros y Arquitectos Consultores, 2004), tienen por finalidad notificar las prohibiciones, restricciones, obligaciones y autorizaciones existentes. En general, tienen la forma de un rectángulo, con el símbolo en forma circular en la parte superior y una leyenda adicional en la parte inferior. El símbolo es de color negro dentro de un círculo de color rojo, que puede indicar prohibición, la leyenda es de color negro y el fondo es blanco.

Señalización vertical informativa. En (APIA XXI Ingenieros y Arquitectos Consultores, 2004), tienen por objeto guiar al usuario con información necesaria sobre identificación de localidades, direcciones, intersecciones, cruces, distancias por recorrer, prestación de servicios, etc. En general, tienen forma rectangular o cuadrada, a excepción de las señales tipo flecha. Las leyendas, símbolos y orlas son de color blanco. El color de fondo de las señales para autopistas es azul y para las vías convencionales, es verde.

Redes Neuronales

Las redes neuronales artificiales son aproximaciones a cómo funciona el cerebro desde una vista matemática, ya que no pueden emular el complejo funcionamiento de una neurona biológica (Cruz, 2010). Zocca, Spacagna, Slate, y Roelants (2017) identifican tres principales características en una red neuronal: la arquitectura de la red neuronal, el entrenamiento y la función de activación. La función de activación es la función matemática que determina la salida de una neurona, recibe como entrada la combinación de los pesos con los valores de entrada y determina si la neurona siguiente debe ser activada o no, además de normalizarla (Ansari, 2020). A continuación, se describe las funciones de activación que se usarán en el desarrollo de la red neuronal:

- **Función sigmoideal o logística.** La función sigmoide es creciente, continua y derivable, es muy útil en redes neuronales que se entrenan con retro propagación (Berzal, 2018). Ansari (2020) indica que la función de activación sigmoide da valores entre 0 y 1, haciendo que la salida sea controlada y tiene la forma:

$$y = f_{sigm}(z) = \frac{1}{1 + e^{-z}}$$

“La importancia de la función sigmoideal es que su derivada siempre es positiva y cercana a cero” (Cruz, 2010, p. 201).

- **Función de activación lineal rectificada (ReLU).** Se trata de una función que no requiere operaciones aritméticas (Berzal, 2018). Ansari (2020) explica que la salida de esta función varía entre 0 (si la entrada es negativa) e ∞ , su ecuación es:

$$y = f_{ReLU}(z) = \max(0, z)$$

En su libro Ansari (2020) indica que las ventajas de esta función es que es computacionalmente eficiente y converge rápido. Es muy usada para modelos de visión artificial.

SoftMax. Es una función que toma un vector de números reales y genera salidas entre 0 y 1 que se interpreta como una probabilidad. Se suele usar en la última capa de una red (Ansari, 2020).

Otro concepto importante en las redes neuronales es el descenso de gradiente, el cual es el método que encuentra el valor mínimo de una función, se usa para minimizar la función de pérdida. Los valores de los coeficientes de las entradas son modificados hasta que la función encuentre el mínimo. La velocidad en que se modifica el valor de los coeficientes es determinada por la tasa de aprendizaje (Abril, s.f.).

Para entender el funcionamiento de una red neuronal, es necesario hacer referencia al perceptrón. Un perceptrón es una sola neurona de una red neuronal que recibe y genera información, su objetivo es hallar los pesos

de cada entrada, a través función de optimización mejora los pesos de cada entrada con un algoritmo, esta optimización es el aprendizaje de la neurona. Su funcionamiento consiste en asignar un peso aleatorio a cada entrada, esta multiplica a el valor de la entrada y la suma de estos genera una salida, siguiendo la ecuación: (Ansari, 2020).

$$f(x) = x_o + \sum_1^i w_i x_i$$

Las limitaciones de la red de una sola capa en la implementación en problemas complejos surgen la necesidad de implementar modelos con capas intermedias, así surge el perceptrón multicapa (Palma Méndez y Marín Morales, 2008). “Los perceptrones multicapa pueden modelar los problemas del mundo real con más precisión” (Ansari, 2020, p. 152). En este modelo se posee por lo menos 3 niveles de neuronas: la capa de entrada, las capas ocultas y la capa de salida. En una MLP una neurona debe estar totalmente conectada a la siguiente capa, las funciones de activación de las capas ocultas son en su mayoría lineal o sigmoideal (Caicedo y López, 2017).

Redes Neuronales Convolucionales

Las redes neuronales convolucionales son muy usadas para el procesamiento de imágenes, ya que permite encontrar características o elementos importantes en las imágenes (Torres, 2020). El problema de una red neuronal totalmente conectada es que el número de neuronas podría ser muy alto por lo cual la idea detrás de las redes neuronales convolucionales es reducir el número de parámetros que permita que una red sea profunda y con menos parámetros (Aghdam y Heravi, 2017).

En este tipo de redes se recibe como entrada un vector cuando la entrada es de 1 dimensión (como por ejemplo un micrófono), una matriz cuando la entrada es de 2 dimensiones (imágenes en blanco y negro) o un tensor si la entrada es mayor a 2 dimensiones (imágenes a color), la salida podría ser convencional como por ejemplo una clasificación con la función *Softmax* o la salida podría ser por ejemplo una imagen que indica a que zona se debe

prestar atención (Berzal, 2018).

Capa convolucional

La capa convolucional aprende patrones locales en una imagen, también aprenden jerarquías espaciales de patrones, ejemplo en una primera capa se puede aprender elementos básicos como aristas y la posterior algún derivado de las aristas (Torres, 2020). Una capa convolucional da conocimiento acerca de características típicas acerca de un problema. A diferencia de una red neuronal multicapa una neurona opera sobre un subconjunto de la entrada con ello cada neurona detecta una característica, si detecta algo útil lo utiliza para buscar algo similar en la siguiente entrada que procesa (Berzal, 2018).

Berzal (2018) indica que la arquitectura de las redes neuronales convolucionales está formada por una secuencia de capas, estas transforman la entrada y generan una salida, casi siempre presentan tres tipos de capas: convolucionales, *pooling* y capas completamente conectadas. Normalmente se combinan capas de convolucionales con capas de *pooling*, a veces se combinan con una capa de tipo *ReLU*. Para realizar una clasificación normalmente se la red finaliza con una capa de tipo *softmax*. Algunos *hiperparámetros* de las redes convolucionales son:

Rellenado con ceros. Se encarga de rellenar con ceros la entrada para obtener un tamaño de salida predeterminado, en el caso de las imágenes, para que tengan el mismo número de píxeles (Berzal, 2018).

Paso, salto o *stride*. “Indica el número de pasos en que se mueve la ventana de los filtros” (Torres, 2020, p. 172). Se puede realizar una convolución saltando posiciones lo que permite reducir el trabajo del computador, por ejemplo, al trabajar con una imagen si realizaría una convolución píxel a píxel (Berzal, 2018).

Capa *Pooling*. Las redes neuronales convolucionales tienen capas de *pooling* que vienen después de una capa de convolución, estas capas realizan

la compresión de la información generada por las capas convolucionales (Torres, 2020). Según Berzal (2018) el uso más común de una capa de *pooling* se basa en combinar píxeles utilizando la función máxima denominado *max pooling*, usada para reducir la dimensionalidad de una imagen. También identifica los dos objetivos de una capa de *pooling*:

- Reduce dimensiones a las entradas.
- Hace que las señales con las que trabajan capas posteriores no varíen en traslaciones en la entrada.

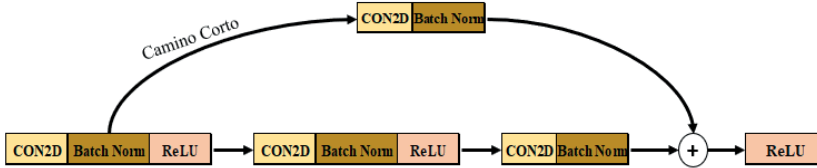
La capa *BatchNormalization*. Tiene por objetivo normalizar las entradas de la capa para que se tenga una activación de salida media de cero y desviación estándar de uno. (Torres, 2020).

La capa *Dropout*. Ayuda a controlar los sobreajustes de modelos, su funcionamiento se basa en no considerar de manera aleatorio en alguna iteración a ciertos conjuntos de neuronas (Torres, 2020).

ResNet (Residual Network)

Las redes tradicionales con muchas capas tienden a sobre ajustarse y son capaces de tener un peor rendimiento, sin embargo, en las *ResNets* mientras más capas se tienen mejor es el rendimiento. Además, las *ResNet* son caracterizadas por su gran rendimiento frente a las otras redes (Venkatesan y Li, 2018). Las redes neuronales residuales o *ResNet* se basan en dividir una capa en dos ramas, una la procesa como cualquier otra red y la otra no la modifica, es decir una rama no modifica el gradiente, lo que permite que una red sea más profunda (Venkatesan y Li, 2018).

Figura 3.
Bloque Residual.



Fuente: (Dwivedi, 2019)

Los saltos en *ResNet* resuelven problemas de la desaparición del descenso de gradiente y garantiza que la capa superior funcione por lo menos mejor que la capa inferior y no peor (Team Great Learning, 2020).

Arquitectura *ResUNet*

Es una red neuronal convolucional que combina la arquitectura *U-Net* con bloques residuales. El uso de bloques residuales permite crear redes neuronales más profundas (Tomar, 2021). Las *U-Net* constan de 3 secciones:

- **Sección de contracción o codificación.** Recibe una imagen y pasa por bloques que reducen la dimensionalidad del mapa de características. (Tomar, 2021). Aplica capa *ReLU* de convolución 3 x 3 y luego una capa de agrupación 2 x 2 (Mittal, 2019).
- **Cuello de botella.** Aplica dos capas de convolución *ReLU* 3 x 3 y concluye con una capa de convolución ascendente de 2 x 2 (Mittal, 2019).
- **Sección de expansión o decodificación.** El decodificador toma el mapa de características y con ello aumenta la imagen y permite generar un mapa de segmentación (Tomar, 2021). Aplica dos capas de convolución *ReLU* 3 x 3 y concluye con una capa de convolución ascendente de 2 x 2, incluye la concatenación con el resultado de las anteriores operaciones (Mittal, 2019).

Con ello *U-Net* tiene una función de pérdida para cada píxel y permite clasificar cada píxel, analizando sus características (Mittal, 2019).

Métodos, Materiales y Herramientas

Las herramientas empleadas fueron:

OpenCV

OpenCV es una biblioteca de visión por computadora, inició como proyecto de Intel, actualmente contiene más de 2.500 algoritmos, también es gratuito ya sea para fines académicos o comerciales. Las utilidades de esta librería pueden ser desde abrir o dibujar formas en imágenes, hasta detectar caras, clasificar imágenes entre muchas otras aplicaciones que se pueden dar. (Marín, 2020).

PIL

Python Image Library es una librería gratuita que permite la edición y procesamiento de imágenes con *Python*, soporta muchos formatos de imágenes. Está diseñada para un acceso rápido los datos, obtener un procesamiento de imágenes eficiente y potente. (Alex Clark and Contributors, s.f.).

Numpy

Numpy es una librería de *Python* con el objetivo en operaciones con un gran volumen de datos, permite representar colecciones de varias dimensiones además que es muy eficiente en la manipulación de datos. Una de sus principales ventajas es que el procesamiento de arreglos es 50 veces más rápido que el de una lista común (Sánchez, 2022).

Jupyter Notebook

Es un software libre de código abierto que permite crear documentos

con código, el cual permite registrar código, ejecutar y ver los resultados. Permite trabajar con bloques de código por separado y soporta *Mark Down*. (Figueiras, 2021). *Jupyter Notebook* tiene dos componentes: Aplicación Web ya que está basada en el navegador y *Notebook documents*, que muestra todo el contenido del cuaderno como las entradas y salidas (texto, imágenes y otras representaciones multimedia) (Team Jupyter, 2015).

Tensorflow

Tensorflow es una biblioteca de código abierto que facilita la creación de modelos de aprendizaje automático. *Tensorflow* ofrece varios niveles de abstracción, permite el entrenamiento e implementación de modelos de forma segura en servidores (Team Tensorflow, s.f.).

Los materiales empleados fueron:

Dataset GTSRB

El *dataset German Traffic Sign Recognition Benchmark (GTSRB)* contiene:

- 51.839 imágenes distribuidas en carpetas para entrenar, validar y realizar las pruebas.
- 43 tipos de señales de tránsito.
- También contiene archivos *.csv* con la información sobre a qué clase corresponde cada imagen.
- Las clases dentro del *dataset* se observan en la tabla 4.

Dataset de Bolivia

La fuente principal de datos es un disco externo proporcionado por el Banco Interamericano de Desarrollo (BID), que cuenta con 1.134 videos de 11.000 kilómetros de la Red Vial Fundamental de Bolivia. Los archivos ocupan alrededor de 386,3 GB con un total de 30.539.674 *frames*. Cada *frame* tiene una resolución de 1280x720 píxeles en cada canal RGB. La tasa de muestreo promedio es de 30,3 FPS. No se tienen sobre la marca de la cámara. Los *frames* se organizan en videos en formato MP4, y los

metadatos, no muestran georreferenciación.

Los datos etiquetados fueron obtenidos por funcionarios, algunas otras imágenes fueron obtenidos por personal asistente, este *dataset* está desbalanceado, es decir algunas señales de tránsito son más prevalentes que otras. El entrenar modelos sobre datos desbalanceados regularmente produce modelos con mucho margen de error. El *dataset* de origen boliviano contiene:

- 9.845 imágenes de carreteras.
- 9.845 archivos en formato *.txt* con información (x, y) de la ubicación de los puntos.
- Los datos del *dataset* son específicamente para implementar en una red neuronal *YOLO*.

Las metodologías empleadas fueron:

Metodología de Investigación

Se realizó con el modelo cuantitativo de tipo experimental basado en el porcentaje de eficacia en la fase de entrenamiento en la clasificación de la señalización vertical con el *dataset* de Bolivia y el *dataset* GTSRB; y en la segmentación con el *dataset* de Bolivia, también al realizar la validación con los datos destinados a la prueba del rendimiento de la red neuronal desarrollada.

Metodología de Desarrollo

Para el desarrollo de la red neuronal se tomará en cuenta una metodología estándar, la cual tiene 4 fases, según (Quispe, s.f.):

- 1. Definición de la red neuronal.** En esta fase se realiza la elección de la arquitectura de la red neuronal. También se toma en cuenta: el tamaño de la red, el tipo de entrada y salida. Esta fase es fundamental ya que tendrá efecto en la eficacia de la red neuronal.

2. **Entrenamiento de la red neuronal.** En esta fase determina los algoritmos de entrenamiento. Es la fase donde la red es capaz de aprender. Como su nombre indica en esta fase se realiza el entrenamiento de la red neuronal.
3. **Utilización de la red neuronal.** En esta fase se brinda una entrada a la red neuronal, a diferencia de la fase de entrenamiento no existe aprendizaje, por lo tanto, no se ajustan los pesos.
4. **Mantenimiento de la red neuronal.** Esta fase sirve para dar un continuo control a la red neuronal, ya sea gracias al aumento del *dataset* o cambios en la arquitectura, ya que los problemas en su mayoría son dinámicos. Cabe resaltar que, por la cantidad de tiempo para el desarrollo del presente artículo, esta fase no será considerada.

Definición de la Red Neuronal

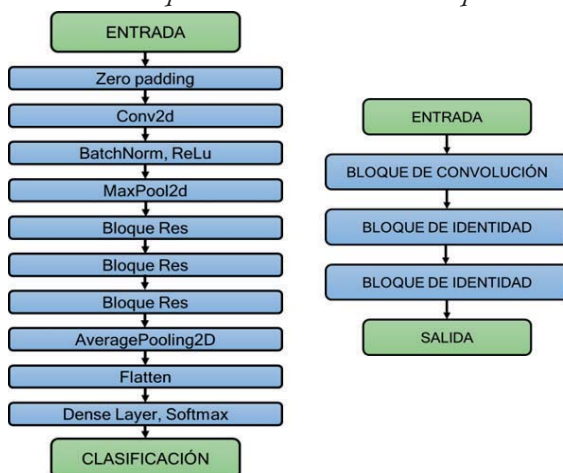
Red Neuronal Nro. 1: Entrenamiento de red neuronal para la clasificación de señales de tránsito usando *GTSRB* y el modelo *ResNet*.

Se realizó la construcción de una red neuronal para la clasificación de señales de tránsito usando como objeto de estudio el *dataset GTSRB*.

- Se consideró la arquitectura de la Figura 4.
- En esta fase ya se cuenta con el *dataset* distribuido en 3 carpetas: “*Meta*”, “*Train*” y “*Test*” con sus respectivos archivos *.csv*, estos archivos tienen la dirección de cada imagen. Nuestro notebook será capaz de leer todas las imágenes de los archivos *.csv*, posteriormente se redimensionará las imágenes a un tamaño 50 x 50 para poder almacenarlo en arreglos.
- La capa de salida usará la función *Softmax* ya que se trata de una tarea de clasificación.

Figura 4

Arquitectura de la red neuronal. Izquierda: Arquitectura de la red neuronal. Derecha: Bloque residual usado en la arquitectura de la red neuronal.



Fuente: Elaboración Propia, modelo basado en (Team Ligncy I, 2021)

Red Neuronal Nro. 1: Entrenamiento de red neuronal para la clasificación de señales de tránsito usando el *dataset* de Bolivia y el modelo *ResNet*.

Se realizó la construcción de una red neuronal para la clasificación de señales de tránsito usando como objeto de estudio el *dataset* de Bolivia. Se generó el *dataset* recortando las señales de tránsito de las imágenes de carreteras y posteriormente se realizó la clasificación manual y la generación del archivo *.csv*.

- Se consideró la arquitectura de la figura 4.

Red Neuronal Nro. 2: Entrenamiento de red neuronal para la detección de señales de tránsito usando el *dataset* de Bolivia y el modelo *ResUNet*.

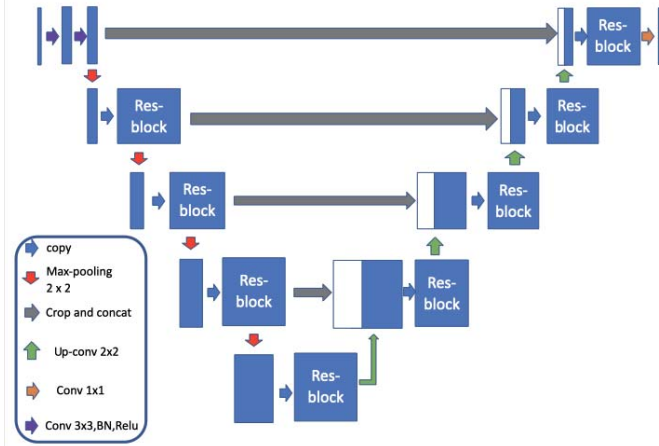
Se realizó la construcción y entrenamiento de la red neuronal para la segmentación y reconocimiento de las señales de tránsito usando como

objeto de estudio el *dataset* de Bolivia que contiene imágenes de carreteras.

- Se utilizó la arquitectura *U-Net* combinada con la *ResNet*, es decir se usó la *ResUNet*, explicada anteriormente.
- En la Figura 5 se muestra la arquitectura.

Figura 5

Arquitectura ResUNet.



Fuente: (Team Ligeny I, 2021)

- En esta ocasión se utilizó el bloque residual especificado en la figura 3.
- El *dataset* actual contiene datos válidos solo para redes neuronales de tipo *YOLO*, con lo cual se realizará la transformación de datos para que pueda ser usado por cualquier tipo de red neuronal.
- A través del nuevo *dataset* se puede generar máscaras usando la librería *PIL* y otras librerías de *Python*, las rutas de estas máscaras serán guardadas en el archivo de extensión *.csv* generado anteriormente.

Entrenamiento de la Red Neuronal

Red Neuronal Nro. 1: Entrenamiento de red neuronal para la clasificación de señales de tránsito usando GTSRB y el modelo *ResNet*.

- La partición de datos es 31367 para el entrenamiento, 768 para la validación y 12630 para la prueba.
- Para la fase de entrenamiento se consideró la configuración de la Tabla 1.

Tabla 1

Configuración para el entrenamiento de la red neuronal nro. 1

Tasa de aprendizaje (<i>Learning rate</i>)	0.001
Épocas (<i>Epochs</i>)	10
Optimizador (<i>Optimizer</i>)	<i>Adam</i>
Función de Pérdida (<i>Loss Function</i>)	<i>categorical_crossentropy</i>
Métricas (<i>Metrics</i>)	<i>accuracy</i>
Tamaño de Entrada (<i>Input Size</i>)	50, 50

Fuente: Elaboración Propia

Red Neuronal Nro. 1: Entrenamiento de red neuronal para la clasificación de señales de tránsito usando el *dataset* de Bolivia y el modelo *ResNet*.

- Para la fase de entrenamiento se consideró la configuración de la Tabla 2.
- La partición de datos es 7797 para el entrenamiento, 1950 para la validación y 2436 para la prueba.

Red Neuronal Nro. 2: Entrenamiento de red neuronal para la detección de señales de tránsito usando el *dataset* de Bolivia y el modelo *ResUNet*.

- La configuración para el entrenamiento se refleja en la Tabla 2.

Tabla 2

Configuración para el entrenamiento para la red neuronal nro. 2.

Tasa de aprendizaje (<i>Learning rate</i>)	0.05
Épocas (<i>Epochs</i>)	7
Optimizador (<i>Optimizer</i>)	Adam
Función de Pérdida (<i>Loss Function</i>)	<i>focal_tversky</i>
Métricas (<i>Metrics</i>)	<i>tversky</i>
Tamaño de Entrada (<i>Input Size</i>)	256, 256

Fuente: Elaboración Propia.

- La partición de datos es 8369 para el entrenamiento, 738 para la validación y 738 para la prueba.
- Al tratarse de un data set no balanceado y no contener un valor muy grande imágenes se usó la función de perdida *focal_tversky*.
- La función de pérdida focal de *Tversky* mejora el equilibrio entre precisión y la segmentación, es importante en problemas desequilibrados y con datos variables, mejorando el rendimiento de la *U-Net*, la redundancia en las características ayuda a recuperar información (Nabila, 2018).

Utilización de la red neuronal

Red Neuronal Nro. 1: Entrenamiento de red neuronal para la clasificación de señales de tránsito usando *GTSRB* y el modelo *ResNet*.

- En la fase de entrenamiento del modelo se obtuvo un 99.08% de eficacia.
- En la fase de evaluación del rendimiento del modelo se obtuvo un 94.36% de eficacia.
- El reporte de clasificación de la red neuronal se ve en la tabla 3.

Tabla 3

Reporte de clasificación de la red neuronal nro. 1 con el dataset GTSRB.

Tipo	Precisión	Soporte	Tipo	Precisión	Soporte
<i>Speed limit (20km/h)</i>	0.97	60	<i>Bumpy road</i>	0.95	120
<i>Speed limit (30km/h)</i>	0.96	720	<i>Slippery road</i>	0.87	150
<i>Speed limit (50km/h)</i>	0.97	750	<i>Road narrows on the right</i>	0.96	90
<i>Speed limit (60km/h)</i>	0.97	450	<i>Road work</i>	0.97	480
<i>Speed limit (70km/h)</i>	0.98	660	<i>Traffic signals</i>	0.91	180
<i>Speed limit (80km/h)</i>	0.90	630	<i>Pedestrians</i>	0.77	60
<i>End of speed limit (80km/h)</i>	0.94	150	<i>Children crossing</i>	0.83	150
<i>Speed limit (100km/h)</i>	0.94	450	<i>Bicycles crossing</i>	0.95	90
<i>Speed limit (120km/h)</i>	0.87	450	<i>Beware of ice/snow</i>	0.77	150
<i>No passing</i>	0.98	480	<i>Wild animals crossing</i>	0.95	270
<i>No passing over 3.5 tons</i>	0.96	660	<i>End speed + passing limits</i>	0.92	60
<i>Right-of-way at intersection</i>	0.80	420	<i>Turn right ahead</i>	1.00	210
<i>Priority road</i>	1.00	690	<i>Turn left ahead</i>	0.92	120
<i>Yield</i>	0.99	720	<i>Ahead only</i>	0.96	390
<i>Stop</i>	1.00	270	<i>Go straight or right</i>	0.95	120
<i>No vehicles</i>	0.99	210	<i>Go straight or left</i>	0.97	60
<i>Veh > 3.5 tons prohibited</i>	1.00	150	<i>Keep right</i>	0.97	690
<i>No entry</i>	0.99	360	<i>Keep left</i>	0.96	90
<i>General Caution</i>	0.93	390	<i>Roundabout mandatory</i>	0.76	90
<i>Dangerous curve left</i>	0.95	60	<i>End of no passing</i>	0.98	60
<i>Dangerous curve right</i>	0.69	90	<i>End no passing veh > 3.5 tons</i>	0.96	60
<i>Double curve</i>	0.95	90			

Fuente: Elaboración Propia

Los resultados se muestran en la figura 6.

Figura 6

Resultados de la red neuronal nro. 1 con el *dataset GTSRB*. **Izquierda:** Resultado de la clasificación de la señal “*General Caution*”. **Centro:** Resultado de la clasificación de la señal “*Veh > 3.5 tons prohibited*”. **Derecha:** Resultado de la clasificación de la señal “*Turn right ahead*”.



Fuente: Elaboración Propia

Red Neuronal Nro. 1: Entrenamiento de red neuronal para la clasificación de señales de tránsito usando el *dataset* de Bolivia y el modelo *ResNet*.

- En la fase de entrenamiento del modelo se obtuvo un 99.55% de eficacia.
- En la fase de evaluación del rendimiento del modelo se obtuvo un 99.55% de eficacia.
- El reporte de clasificación de la red neuronal se ve en la tabla 4.

Tabla 4

Reporte de clasificación de la red neuronal nro. 1 con el dataset de Bolivia.

Tipo	Prec.	Sop.	Tipo	Prec.	Sop.
Altura máxima permitida 5.5m	0.67	41	Intersección con camino lateral a la oblicuo a la izquierda	1.00	12
			Intersección con camino lateral a la oblicuo a la izquierda baja		
Bajada peligrosa	1.00	20	Intersección con camino lateral a la oblicuo a la izquierda baja	0.89	8

Detección de la señalización de tránsito vertical con redes neuronales convolucionales basadas en bloques residuales

Ceda el paso	1.00	7	Intersección con caminos laterales izquierda derecha	0.74	14
Centro de salud	0.59	22	Intersección rotatoria	0.61	25
Chevron a la derecha	1.00	125	Loma	1.00	112
Chevron a la izquierda	1.00	313	No usar el celular	0.00	2
Ciclistas	1.00	5	Parada de transporte	1.00	7
Comienzo del camino dividido	1.00	14	Pare	1.00	16
Cruce	0.62	44	Paso a nivel sin barrera	0.60	7
Curva cerrada a la izquierda	0.00	17	Paso peatonal	1.00	90
Curva pronunciada a la derecha	0.71	141	Paso sobre nivel	1.00	13
Curva pronunciada a la izquierda	0.90	100	Peatones	1.00	11
Curva y contracurva cerrada primera a la izquierda	0.69	30	Permitido estacionar	0.00	18
Curva y contracurva pronunciada primera a la derecha	1.00	9	Prohibido adelantar	1.00	100
Curva y contracurva pronunciada primera a la izquierda	1.00	8	Prohibido capturar animales	0.00	6
Estación de servicio	1.00	42	Prohibido prender fuego	0.00	3
Estrechamiento del camino en el lado derecho	1.00	12	Retorno	1.00	6
Estrechamiento del camino en el lado izquierdo	1.00	77	Tránsito pesado a la derecha	1.00	20
Final del camino dividido	0.10	13	Turril	1.00	21

Flechas	0.99	110	Velocidad máxima 30km/h	0.00	22
Ganado suelto	0.98	91	Velocidad máxima 35km/h	0.00	9
Giro en U a la derecha	1.00	3	Velocidad máxima 40km/h	0.40	12
Giro en U a la izquierda	1.00	15	Velocidad máxima 50km/h	0.17	25
Iglesia	0.00	6	Velocidad máxima 60km/h	0.46	35
Incorporación de tránsito por lado izquierdo	1.00	14	Velocidad máxima 70km/h	0.43	39
Intersección con camino lateral a la derecha	0.96	194	Velocidad máxima 80km/h	0.00	5
Intersección con camino lateral a la izquierda	0.16	26	Velocidad máxima 100km/h	0.28	67
Intersección con camino lateral a la oblicuo a la derecha	0.83	54	Zona de curvas sucesivas a la primera a la izquierda	0.73	34
Intersección con camino lateral a la oblicuo a la derecha baja	0.58	23	Zona de curvas sucesivas a la primera a la derecha	1.00	65
			Zona escolar	0.52	56

Fuente: Elaboración Propia

- Los resultados se muestran en la Figura 7.

Figura 7

Resultados de la red neuronal nro. 1 con el dataset de Bolivia.

Izquierda: Resultado de la clasificación de la señal “Chevron a la izquierda”. **Centro:** Resultado de la clasificación de la señal “Intersección con camino lateral a la derecha”. **Derecha:** Resultado de la clasificación de la señal “Final de camino dividido”.



Predicción: Chevron a la izquierda

Verdadero: Chevron a la izquierda



Predicción: Intersección con camino lateral a la derecha

Verdadero: Intersección con camino lateral a la derecha



Predicción: Final de camino dividido

Verdadero: Final de camino dividido

Fuente: Elaboración Propia

Red Neuronal Nro. 2: Entrenamiento de red neuronal para la detección de señales de tránsito usando el dataset de Bolivia y el modelo ResUNet.

-
- En la fase de entrenamiento del modelo se obtuvo un 75.83% de eficacia.
- El resultado se puede observar en la Figura 8.

Figura 8.

Resultados de la red neuronal nro. 2 con el dataset de Bolivia.

Izquierda: Imagen original de la carretera. **Centro:** Máscara original para la segmentación. **Derecha:** Máscara generada por la red neuronal.



Fuente: Elaboración Propia

Red Neuronal Nro. 1 y Red Neuronal Nro. 2: Segmentación y clasificación de las señales de tránsito usando videos

- La combinación de ambas redes neuronales ayuda a la detección de las señales de tránsito en primera instancia, a través de la máscara para posteriormente poder clasificar la zona detectada.
- Se aplicó erosión y dilatación (ambos muy usados en visión artificial) a la máscara para perfeccionar esta y eliminar el ruido.
- La librería *OpenCV* fue de vital importancia ya que aportó las funciones para la lectura/escritura de video, detectar el contorno de las imágenes, erosión, dilatación, entre muchas otras
- En la Figura 9 se puede observar los resultados de la combinación de ambas redes neuronales.

Figura 9.

Resultado de la red neuronal nro. 1 y red neuronal nro. 2 combinadas.

Arriba: Resultado de la segmentación y clasificación sin error. **Abajo:**

Resultado de la segmentación y clasificación con error en la clasificación



Fuente: Elaboración Propia

Discusión

Si bien el resultado de la red neuronal nro. 1 con el *dataset* de Bolivia es alto con un valor de eficacia de un 99.55%, los valores obtenidos en el reporte de clasificación de la red neuronal usando el *dataset* de Bolivia muestran errores al momento de clasificar ciertas clases, por ejemplo: “*Curva cerrada a la izquierda*”, “*Prohibido prender fuego*”, “*No usar el celular*”, entre otros, los cuales tienen un valor de precisión de 0%, esto se debe a la cantidad reducida de imágenes para ciertas clases ya que se cuenta con un *dataset* desbalanceado y no pudo realizar la validación de estas clases, es decir fueron omitidos. Es necesario incrementar las imágenes del *dataset* de Bolivia para mejorar la clasificación.

Con el *dataset* *GTSRB* la red neuronal tuvo un porcentaje alto ya que cuenta con un *dataset* balanceado, sin embargo, la desventaja de este *dataset* radica en el origen, ya que no puede ser usado en nuestro entorno por los temas del idioma además de ciertas diferencias en los colores en la señalización vertical.

Conclusiones y Recomendaciones

La combinación de la red neuronal nro. 1 y red neuronal nro. 2 permiten la detección y clasificación de señales de tránsito, esta combinación puede ser usada de diferentes y variadas maneras para poder reducir el número de accidentes en Bolivia.

El porcentaje de eficacia de la red neuronal nro. 1 con el *dataset* *GTSRB* es alto, obteniendo un valor de 94.36%, además incluye valores alto en el reporte de clasificación, caso contrario sucede con el *dataset* de Bolivia debido a que el *dataset* está desbalanceado.

La clasificación de señales de tránsito de Bolivia aún contiene errores en la predicción, por lo cual se requiere aumentar el *dataset* de Bolivia, es decir entrar a la fase de mantenimiento.

Las redes neuronales convolucionales en general tienen un gran rendimiento en el procesamiento de imágenes, sin embargo, las arquitecturas de redes neuronales que involucran bloques residuales como las *ResNet* y *ResUNet*, evitan la desaparición del descenso de gradiente lo que permite trabajar con redes neuronales más profundas lo que permite mayor precisión en los resultados.

Se recomienda utilizar otras arquitecturas de redes neuronales para realizar la detección y clasificación de señales de tránsito, como por ejemplo la red neuronal *YOLO V4*.

Referencias

- Abril, R. (s.f.). *El descenso del gradiente*. La máquina oráculo. <https://lamaquinaoraculo.com/computacion/el-descenso-del-gradiente/>
- Aghdam, H., & Heravi, E. (2017). *Guide to Convolutional*. Editorial: Springer International. Obtenido de <https://www.amazon.com/Guide-Convolutional-Neural-Networks-Classification/dp/331957549X>
- Ansari, S. (2020). *Building Computer Vision Applications Using Artificial Neuronal Networks*. Virginia, Editorial: Apress. Retrieved from <https://www.amazon.com/-/es/Shamshad-Ansari/dp/148425886X>
- APIA XXI Ingenieros y Arquitectos Consultores. (2004). *Manual de Dispositivos de Control de Tránsito*. Administradora Boliviana de Carreteras. Obtenido de https://www.abc.gob.bo/wp-content/uploads/2018/09/manual_de_dispositivo_de_control_de_transito.pdf
- Automóvil Club Boliviano. (2004). *Educación y Seguridad Vial*. Automóvil Club Boliviano. Editorial: Automóvil Club Boliviano.
- Berzal, F. (2018). *Redes Neuronales & Deep Learning*. Granada. Obtenido de <https://vdoc.pub/documents/redes-neuronales-deep-learning-4ou7rq5ahv80>
- Caicedo, E., & López, J. (2017). *Una aproximación práctica a las redes neuronales artificiales*. Editorial: Universidad del Valle.

- Obtenido de <https://bibliotecadigital.univalle.edu.co/bitstream/handle/10893/10330/Una-aproximacion-practica-a-las-redes.pdf?sequence=6>
- Camacho, F., y Labrandero, J. (2016). “Redes neuronales convolucionales aplicadas a la traducción del lenguaje verbal español al lenguaje de señas Boliviano”. *Revista Ciencia, Tecnología e Innovación*. 12(13) 755-762 http://www.scielo.org.bo/pdf/rcti/v12n13/v12n13_a05.pdf
- Charniak, E. (2018). *Introduction to Deep Learning*. Editorial: The MIT Press. Retrieved from <https://www.amazon.com/Introduction-Deep-Learning-MIT-Press/dp/0262039516>
- Clark, C and Contributors. (s.f.). *Pillow*. *Pillow*. Obtenido de <https://pillow.readthedocs.io/en/stable/>
- Cruz, P. (2010). *Inteligencia Artificial con aplicaciones a la ingeniería* (1ra ed.). Editorial: Alfaomega. Obtenido de <https://lelinopontes.files.wordpress.com/2014/09/inteligencia-artificial-con-aplicaciones-a-la-ingenieria3ada.pdf>
- Dwivedi, P. (4 de Enero de 2019). *Understanding and Coding a ResNet in Keras*. Obtenido de Towards Data Science: <https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33>
- Figueiras, S. (20 de Septiembre de 2021). *¿CONOCES JUPYTER NOTEBOOK?*. Centro Europeo de Postgrado y Empresa. <https://www.ceupe.mx/blog/conoces-jupyter-notebook.html>
- Instituto Nacional de Estadística. (2020). *REGISTROS ADMINISTRATIVOS*. Obtenido de HECHOS DE TRÁNSITO: <https://www.ine.gob.bo/index.php/registros-administrativos-seguridad/>
- Marín, R. (12 de Febrero de 2020). *¿Qué es OpenCV? Instalación en Python y ejemplos básicos*. *Revista Digital INESEM*. <https://www.inesem.es/revistadigital/informatica-y-tics/opencv/>
- Mittal, A. (3 de Junio de 2019). *Introduction to U-Net and Res-Net for Image Segmentation*. Medium. <https://aditi-mittal.medium.com/introduction-to-u-net-and-res-net-for-image-segmentation-9afcb432ee2f>

- Nabila, N. (2018). A novel Focal Tversky loss function with improved Attention U-Net for lesion segmentation. *arXiv preprint arXiv:1810.07842*. Obtenido de <https://arxiv.org/abs/1804.03999>
- Palma, J., & Marín, R. (2008). *Inteligencia Artificial: Métodos, técnicas y aplicaciones*. Editorial: McGraw-Hill. Obtenido de https://www.ingebook.com/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=4085
- Quispe, N. (s.f.). *Redes Neuronales Artificiales, Metodología de Desarrollo y Aplicaciones*. <https://www.monografias.com/trabajos95/redes-neuronales-artificiales-metodologia-desarrollo-y-aplicaciones>
- Reynaga, R., & Yupanqui, F. (2022). *Inteligencia Artificial Aplicada a Proyectos de Transporte*. Redes Neuronales para la Clasificación de Señales de Tránsito. La Paz.
- Sánchez, A. (12 de Mayo de 2022). *La librería Numpy*. Aprende con Alf. <https://aprendeconalf.es/docencia/python/manual/numpy/>
- Team Great Learning. (28 de Septiembre de 2020). *Introduction to Resnet or Residual Network*. Great Learning. <https://www.mygreatlearning.com/blog/resnet/#sh12>
- Team Jupyter. (2015). *Jupyter Notebook*. Jupyter Notebook. <https://jupyter-notebook.readthedocs.io/en/stable/notebook.html>
- Team Ligency I. (Noviembre de 2021). *Masterclass en Inteligencia Artificial: con Proyectos Reales*. Obtenido de Udeemy: <https://www.udemy.com/course/masterclass-en-inteligencia-artificial-crea-6-proyectos/>
- Team Tensorflow. (s.f.). *Tensorflow Core*. Tensorflow. <https://www.tensorflow.org/overview?hl=es-419>
- Tomar, N. (6 de Febrero de 2021). *What is RESUNET*. Idiot Developer: <https://idiotdeveloper.com/what-is-resunet/>
- Torres, J. (2020). *Python Deep Learning. Introducción Práctica con Keras y TensorFlow 2* (1ra ed.). Editorial: Alfaomega. Obtenido de <https://www.amazon.com/-/es/Jordi-Torres/dp/8426728286>
- Valero, J., Zúñiga, A., & Clares, J. (2021). “Detección de la tuberculosis con algoritmos de Deep Learning en imágenes de radiografías del tórax”. . Revista de Investigación en Salud. 4(12) 624-633 <http://www.scielo.org.bo/pdf/vrs/v4n12/2664-3243-vrs-4-12-190.pdf>

- Venkatesan, R., & Li, B. (2018). *Convolutional Neural Networks in Visual Computing A Consice Guide*. Editorial: CRC Press. Retrieved from <https://www.amazon.com/-/es/Ragav-Venkatesan-ebook/dp/B076PPMR3C>
- Zocca, V., Spacagna, G., Slate, D., & Roelants, P. (2017). *Python Deep Learning*. Editorial: Packt. Obtenido de <https://www.amazon.com/Python-Deep-Learning-techniques-architectures-ebook/dp/B07KQ29CQ3>

