

Reconocimiento de Señales de Tráfico Alemanas Mediante Redes Neuronales Convolucionales Utilizando el Conjunto de Datos GTSRB

Sara Lorena Duque Ramírez
Ingeniería de Sistemas, Universidad de Antioquia
Medellín, Colombia
sara.duque1@udea.edu.co

Resumen— Se analiza el reconocimiento de señales de tráfico alemanas utilizando redes neuronales convolucionales (CNN) y el conjunto de datos GTSRB. Se describe la estructura de varios notebooks que abordan la exploración, preprocesamiento y arquitectura del dataset. En la primera parte, se realiza una exploración descriptiva del conjunto de datos, donde se identifican y corrigen anomalías como imágenes borrosas o negras. Posteriormente, se define la arquitectura de la CNN, que incluye capas convolucionales para extracción de características, capas de pooling para reducción dimensional, y capas densas para la clasificación final. El modelo se entrena durante 10 épocas con un conjunto de 7084 ejemplos, logrando una precisión del 97.79% en la clasificación de señales. Las métricas de desempeño como precisión, recall y F1-score también indican un rendimiento excelente.

I. DESCRIPCIÓN DE LA ESTRUCTURA DE LOS NOTEBOOKS ENTREGADOS

Los notebooks presentados abordan el análisis, preprocesamiento y modelado de un dataset de imágenes de tránsito. Incluyen secciones para explorar y visualizar datos, detectar y corregir anomalías, preprocesar imágenes, definir una arquitectura basada en redes neuronales convolucionales (CNN), y evaluar el desempeño del modelo mediante métricas y visualización de resultados.

A. Estructura del Notebook "Exploración del dataset"

El notebook está diseñado para la exploración y análisis de un conjunto de datos de imágenes, dividiéndose en secciones organizadas para abordar aspectos específicos del preprocesamiento y evaluación de calidad de los datos. A continuación, se describe su estructura:

1) Introducción y configuración inicial: Incluye

celdas dedicadas a la importación de librerías necesarias, y a la configuración del entorno, el montaje de Google Drive para acceder al dataset. También se establecen las rutas y directorios clave para organizar los datos.

2) *Exploración Descriptiva del Dataset:* Contiene celdas para la inspección de los contenidos del dataset, como la enumeración de carpetas y archivos en directorios específicos (carpetas de entrenamiento y prueba).

Introduce una clasificación inicial, definiendo categorías y etiquetas relacionadas con las imágenes del dataset.

3) *Visualización y Resumen de Datos:* Proporciona funciones que generan resúmenes cuantitativos del dataset, como el número de imágenes por categoría y su distribución.

Integra métodos para visualizar ejemplos de imágenes por clase, lo que facilita una inspección visual directa de los datos.

4) *Análisis de Calidad del Dataset:* Define y ejecuta funciones para detectar anomalías comunes en los datos, tales como imágenes negras, borrosas o corruptas. Estas funciones incluyen métodos automatizados que identifican y cuantifican dichas irregularidades.

5) *Preprocesamiento de Imágenes:* Aunque no se observa un procesamiento avanzado en esta etapa, el notebook prepara el terreno para realizar ajustes o depuraciones basadas en los problemas detectados durante el análisis.

B. Estructura del Notebook "Preprocesado del dataset"

En el notebook se preprocesa el dataset importando librerías y montando Google Drive para acceder a los

datos. Se definen los directorios y se listan y leen los archivos CSV. Se resumen y muestran las imágenes de entrenamiento. Se detectan y eliminan imágenes negras y borrosas para mantener la calidad del dataset. Estos pasos son importantes para poder entrenar los modelos de manera eficiente.

1) *Introducción y configuración inicial*: Se importan las librerías necesarias que serán usadas para la manipulación de datos, visualización y procesamiento de imágenes y se monta Google Drive para acceder a los datos del dataset de manera eficiente.

2) *Gestión del dataset*: Se define el directorio raíz del dataset, permite ubicar todos los archivos y también se tiene la lectura de los archivos CSV (estos archivos contienen los datos estructurados).

3) *Visualización y limpieza de imágenes*: Se detectan y eliminan imágenes negras que son muy oscuras, probablemente por errores cuando se tomó la foto o problemas de iluminación, así mismo se detectaron y eliminaron imágenes borrosas que carecen de nitidez.

C. Estructura del notebook “Definición de la Arquitectura”

El notebook está estructurado en varias secciones que abarcan desde la preparación de datos hasta la evaluación y visualización de los resultados del modelo de clasificación basado en redes neuronales convolucionales (CNN).

1) *Importación de librerías*: Se importan las librerías esenciales para la manipulación de datos, procesamiento de imágenes, construcción y evaluación del modelo. Incluye módulos de TensorFlow, sklearn, matplotlib, PIL.

2) *Montaje y Definición del Dataset*: Se monta Google Drive para acceder al dataset ubicado en la nube, además se define la estructura del dataset con rutas específicas para entrenamiento y prueba.

3) *Preparación de generadores de datos*: Se utilizan generadores de imágenes para aumentar los datos, dividir el conjunto de entrenamiento en entrenamiento y validación, cargar imágenes desde directorios.

4) *Definición del modelo*: Se implementa la red neuronal convolucional (CNN) compuesta por capas convolucionales, capas de agrupamiento, normalización por lotes, regularización y capa densa.

5) *Configuración y entrenamiento del modelo*: Se define un optimizador, se compila el modelo, además

se entrena el modelo durante 10 épocas, visualizando la pérdida (loss) y la precisión (accuracy).

6) *Evaluación del modelo*: Se evalúa el desempeño del modelo tanto en el conjunto de entrenamiento como en el de validación.

7) *Carga y Preprocesamiento de Imágenes de Prueba*: Se carga el archivo CSV con etiquetas y rutas de imágenes de prueba, las imágenes se procesan para redimensionarlas y normalizarlas.

8) *Predicción y métricas de desempeño*: Se realizan predicciones sobre el conjunto de datos de prueba, se calculan métricas como accuracy, precision, recall, F1-Score y matriz de confusión.

9) *Visualización de predicciones*: Se muestran imágenes de validación con sus etiquetas reales y etiquetas predichas.

II. DESCRIPCIÓN DE LA SOLUCIÓN

Aquí se describe la solución planteada para clasificar señales de tránsito con una red neuronal convolucional (CNN). La CNN consta de varias capas, incluyendo capas convolucionales para la extracción de características, capas de pooling para reducir la dimensión espacial, y capas de normalización y dropout para estabilizar el entrenamiento y prevenir el sobreajuste. Después de estas capas, las características se transforman en un vector unidimensional mediante la capa de flatten, y finalmente, las capas densas conectan las características procesadas con las clases de salida, utilizando una función de activación Softmax.

A. Arquitectura de la Red Neuronal Convolucional (CNN)

La arquitectura propuesta es una red neuronal convolucional (CNN) diseñada para clasificar señales de tránsito. A continuación, se describen detalladamente las capas y los elementos principales de la arquitectura:

La CNN sigue un diseño secuencial, donde cada capa tiene un propósito específico, desde la extracción de características hasta la clasificación. La entrada de la red tiene dimensiones fijas de $50 \times 50 \times 3$ (altura, ancho y canales de color).

1) *Capas convolucionales*: Las capas convolucionales se utilizan para extraer características relevantes de las imágenes de entrada, se tienen 4 capas convolucionales.

TABLA I
CONFIGURACIÓN DE LAS CAPAS
CONVOLUCIONALES DE LA CNN

Esta tabla se organizan los parámetros esenciales de cada capa convolucional utilizada en el modelo, incluyendo el número de filtros, tamaño del kernel, función de activación y las dimensiones de entrada de los datos.

Capa convolucional	Filtros	Tamaño del kernel	Activación	Entrada
Primera	16	5 x 5	ReLU	Imagen RGB (50x50x3)
Segunda	32	5 x 5	ReLU	
Tercera	64	3 x 3	ReLU	
Cuarta	64	3 x 3	ReLU	

2) *Capas de Pooling*: Las capas de pooling maximizan la eficiencia al reducir la dimensión espacial, manteniendo las características importantes.

TABLA II
CONFIGURACIÓN DE LAS CAPAS POOLING DE LA CCN

Esta tabla se detalla las características y el orden de las capas de Max Pooling empleadas en el modelo, incluyendo el tamaño del pool y la ubicación dentro de la arquitectura (después de las capas convolucionales correspondientes).

Capa Pooling	Tamaño del pool	Aplicación
Primera	2 x 2	Después de la segunda capa convolucional
Segunda	2 x 2	Después de la cuarta capa convolucional

3) *Normalización por batch (batch normalization)*: Se incluyen capas de normalización para estabilizar y acelerar el entrenamiento. La primera normalización es aplicada después de la primera capa de Max Pooling y la segunda normalización es aplicada después de la segunda capa de Max Pooling.

4) *Capa de dropout*: Se introducen capas de dropout para reducir el riesgo de sobreajuste. Se tiene una capa de dropout con una tasa de eliminación de 25% y es aplicada después de la segunda normalización por batch.

5) *Capa de flatten*: Transforma las características extraídas en un vector unidimensional para conectarlas con las capas densas.

6) *Capas Densas*: Conectan las características procesadas a las clases de salida. En la primera capa densa contiene 512 neuronas, la función de activación es ReLU e incluye normalización por batch y dropout de 25%, la capa de salida contiene 43 neuronas correspondiente a las 43 clases de señales de tránsito, la función de activación es Softmax que genera una distribución de probabilidad para cada clase.

III. DESCRIPCIÓN DE LAS ITERACIONES

El modelo se entrenó utilizando un conjunto de datos de entrenamiento compuesto por 7084 ejemplos y un tamaño de lote batch_size=4. El proceso de entrenamiento se llevó a cabo durante 10 épocas. Cada época consistió en 1771 pasos de entrenamiento, que correspondieron a la cantidad de lotes procesados, que se calcula dividiendo el tamaño del conjunto de datos de entrenamiento que es igual a 7084 sobre el tamaño del lote que es igual a 4 y el número total de pasos sería igual a los pasos por época por el número de épocas esto es igual a 17710.

TABLA III
DETALLES DEL PROCESO DE
ENTRENAMIENTO POR ÉPOCA

Esta tabla muestra los resultados del proceso de entrenamiento y validación del modelo a lo largo de las épocas. Para cada época, se incluyen la precisión y la pérdida tanto en el conjunto de entrenamiento como en el conjunto de validación.

Epoch	accuracy	loss	val_accuracy	val_loss
1	0.5666	1.7336	0.9376	0.2124
2	0.9383	0.2353	0.9509	0.1748
3	0.9648	0.1299	0.9681	0.1108
4	0.9759	0.0898	0.9655	0.1116
5	0.9811	0.0699	0.9757	0.0813
6	0.9844	0.0595	0.9680	0.1013
7	0.9866	0.0491	0.9742	0.0969
8	0.9890	0.0405	0.9707	0.1111
9	0.9880	0.0438	0.9773	0.0915
10	0.9889	0.0390	0.9850	0.0662

En cada época, el modelo realizó un total de 1771 iteraciones, procesando un lote de 4 ejemplos en cada una. En cada iteración, el modelo ejecutó los siguientes pasos:

1) *Propagación hacia adelante (forward pass)*: Los 4 ejemplos del lote pasaron a través de la red neuronal, generando predicciones.

2) *Cálculo de la pérdida (loss)*: La pérdida se calculó en función de la diferencia entre las predicciones del modelo y los valores reales de las etiquetas de

entrenamiento.

3) *Retropropagación (backpropagation)*: El gradiente de la pérdida se calculó con respecto a los pesos del modelo y se usó para actualizar los parámetros del modelo a través de un proceso de optimización, se utilizó el optimizador Adam.

4) *Actualización de pesos*: Los pesos del modelo se actualizaron utilizando el algoritmo de optimización con el objetivo de minimizar la pérdida.

IV. DESCRIPCIÓN DE LOS RESULTADOS.

Se presenta el rendimiento y las métricas de desempeño del modelo entrenado para identificar señales de tránsito. Se analizan aspectos como la precisión, recall, exactitud, F1 score y matriz de confusión.

A. Rendimiento del modelo

El modelo ha sido entrenado adecuadamente, pero podría necesitar ajustes para mejorar su capacidad de generalización y evitar el sobreajuste. Utilizar más datos de entrenamiento, o ajustes en la arquitectura del modelo.

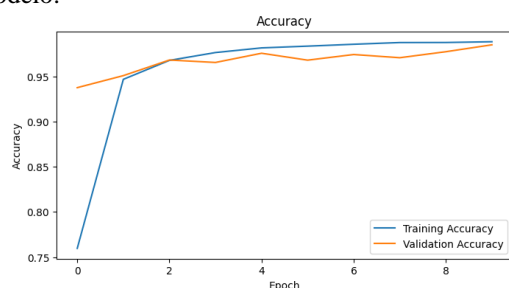


Fig 1. Evolución de la precisión del modelo durante el entrenamiento y la validación

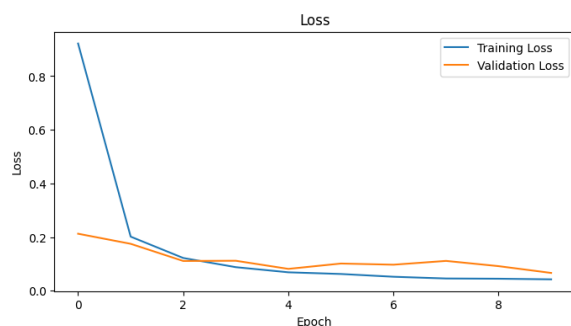


Fig 2. Evolución de la pérdida del modelo durante el entrenamiento y la validación

Después de la compilación del modelo y de la evaluación del modelo en un conjunto de datos de entrenamiento y validación se obtuvieron los siguientes valores:

- Train Loss: 0.00029041568632237613
- Train Accuracy: 0.999964714050293
- Validation Loss: 0.06623019278049469
- Validation Accuracy: 0.9849943518638611

B. Métricas de desempeño del modelo

Estas métricas reflejan que el modelo tiene un desempeño excelente, con un alto grado de precisión, recall y exactitud, lo que indica que está clasificado correctamente la mayoría de las instancias, tanto positivas como negativas.

1) *Exactitud (Accuracy)*: Es la proporción de predicciones correctas sobre el total de predicciones realizadas. En este caso, una exactitud del 97.79% indica que el modelo clasificó correctamente el 97.79% de los ejemplos en el conjunto de datos.

2) *Precisión (Precision)*: Mide la proporción de verdaderos positivos (predicciones correctas de la clase positiva) sobre todas las predicciones positivas realizadas. Un valor de 97.81% significa que el 97.81% de las veces que el modelo predijo la clase positiva, esta fue correcta.

3) *Recall (Sensibilidad)*: Indica la capacidad del modelo para identificar todos los verdaderos positivos, es decir, la proporción de verdaderos positivos sobre el total de casos reales positivos. Un recall de 97.79% significa que el modelo detectó correctamente el 97.79% de todos los casos positivos en el conjunto de datos.

4) *F1 Score*: Es la media armónica entre la precisión y el recall, lo que lo convierte en una métrica que equilibra ambos aspectos. Un F1 score de 97.77% indica que el modelo tiene un buen balance entre precisión y recall, siendo muy eficiente tanto en evitar falsos positivos como en detectar la mayor parte de los verdaderos positivos.

5) *Matriz de confusión*: La mayoría de las señales han sido clasificadas correctamente, como lo indican los valores en la diagonal principal.

