

IMPLEMENTACION DE UN MODELO DE DEEP LEARNING (PORTAFOLIO DE IMPLEMENTACIÓN)

Kevin Joan Delgado Pérez – A01706328

Resumen

Se pretende analizar y documentar el proceso de desarrollo de un modelo de Deep learning, en este caso, el entrenamiento de una red neuronal para la detección de 4 tipos de señales de tráfico.

Introducción

El dataset a utilizar inicialmente contiene distintas señales de tránsito (43 señales), para generar un buen modelo adaptable a las 4 señales como objetivo, se pretenderá usar Transfer learning, el cual consiste en entrenar un modelo con un dataset inicial de reconocimiento de imágenes, para ser vuelto a entrenarse con un distinto dataset, para generar la adaptabilidad al último mencionado. Para comprobar el uso de datos reales, se pretende realizar pruebas mediante visión artificial, detectando señales de tránsito a través de la cámara de una computadora.

Descripción del método

Para implementar el método primero es necesario extraer el set de datos de acuerdo a las carpetas de entrenamiento que se tienen, en este caso, el primer modelo contiene carpetas de acuerdo a la categoría de señal de tránsito, al tener 43 carpetas, se tendrán 43 categorías de entrenamiento, el proceso de entrenamiento consiste en la categorización de clases, el Split de train-test, la construcción del modelo de red neuronal. Como parte importante a mencionar, se consideró el categorical_crossentropy para encontrar la pérdida y evaluar el modelo. Finalmente, para el dataset a entrenar, el resultado fue extraordinario, pues se llegó 97% de precisión del modelo al evaluar distintas imágenes encontradas en su carpeta de prueba.

Sin embargo, para mejorar la adaptabilidad del modelo con la prueba de la cámara, el modelo realizará transfer learning con un dataset más pequeño, al realizar una serie de pruebas iniciales, se encontró que esto daría problemas de flexibilidad en cuanto a algún otro dataset, por lo que se optó por usar data augmentation. También es importante mencionar que fueron congeladas las primeras capas de neuronas, representando la mitad de los parámetros totales.

Terminando de entrenar el modelo, se obtuvieron resultados de una precisión cercana a 45%, se guardó el modelo y se hicieron pruebas de validación manual, realizando un método de detección de círculos, recortando un “marco” de la imagen para mejorar la detección de señales de tránsito, la dinámica de la cámara funcionará igual. Finalmente se guardó el modelo.

Mejoramiento del modelo

Para el mejoramiento del modelo, se procedió a cambiar hiperparámetros tales como el porcentaje de Split, los valores de data augmentation, las capas de neuronas a congelar, puesto que al usar transfer learning es preferible entrenar el menor número de capas de neuronas posibles, al descongelar más, podría provocar una disminución en la efectividad del modelo. Algunas otras modificaciones fueron, agregar capas de dropout y disminuir el tamaño del batch para ralentizar el procesamiento de las imágenes, lo cual provoca mayor minuciosidad a la hora de entrenar el modelo. Esto provocó un mejoramiento en la precisión del modelo, aumentando a un rango de 52% hasta un 61% de precisión.

Resultados

Finalmente, como parte de los resultados se generó un video explicando el proceso de entrenamiento, los archivos generados, y la dinámica de pruebas mediante la cámara y la detección de señales de tránsito, la efectividad fue mejor de la esperada según la precisión obtenida. Algo importante a mencionar fue que se buscó usar modelos pre-entrenados como VGG16, sin

embargo, los resultados de precisión fueron similares o peores que los propuestos en este documento.

Link del video:

https://drive.google.com/file/d/1dw2shz6J53aCbvsryGoKV7blvdd5RGQy/view?usp=share_link

Referencias

Owner Mykola, User from Kaggle (2018). *GTSRB - German Traffic Sign Recognition Benchmark – License Public Domain*. From Kaggle Datasets. Recuperado el 19 de Noviembre de 2022 de:

<https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign>