

Modelo de Reconocimiento Automático de Señales de Tránsito Vehicular mediante Aprendizaje Profundo de Redes Neuronales Convolucionales

Josué Gastón Távara Idrogo
jtavara@unitru.edu.pe

Universidad Nacional de Trujillo
Facultad de Ciencias Físicas y Matemáticas
Escuela Académico Profesional de Informática

Defensa de tesis 03/07/2019



Resumen

La presente investigación tiene por objetivo principal implementar un modelo basado en el aprendizaje profundo de redes neuronales convolucionales para reconocer automáticamente señales de tránsito vehicular usando técnicas de procesamiento de imágenes y fundamentos de inteligencia artificial.

El proyecto se centra en un grupo de señales de Tránsito vehicular de Alemania y Perú, identificando 43 y 7 categorías respectivamente. Iniciando con la adquisición de imágenes, se procedió realizar el procesamiento de estas con la finalidad de aumentar el conjunto de datos y poder ejecutar el aprendizaje profundo a través de diversos diseños de arquitecturas de redes neuronales convolucionales.

Como resultado final, se obtuvo un modelo con buenos indicadores y resultados en el reconocimiento de señales de tránsito vehicular. De esta manera, se pretende contribuir en los esfuerzos de la industria automotriz en el campo de sistemas avanzados de asistencia al conductor así como también puede formar parte de diversos mecanismos que buscan dar soluciones a la inseguridad vial.

Palabras claves: aprendizaje profundo, redes neuronales convolucionales, procesamiento de imágenes.



Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

1 Introducción

2 Motivación

3 Formulación del problema e Hipótesis

4 Importancia de la investigación

5 Contribución de la investigación

6 Marco teórico

- Aprendizaje Profundo
- Red Convolutacional
- Arquitectura del Modelo
- Componentes del Modelo
- Entrenamiento y Validación

7 Propuesta o tema central de la tesis

8 Resultados de la tesis

9 Consideraciones finales

10 Referencias bibliograficas



Introducción

- Al conducir en pistas o carreteras, a veces es difícil mantener los ojos en todas partes a la vez, comprobando el camino por delante, hacia donde girar, dónde disminuir la velocidad o tratar de mantenerla y no acelerar; es por ello, que existen mecanismos destinados a reglamentar el tránsito, advertir o informar a los usuarios mediante palabras, sonidos o símbolos determinados.



Introducción

- Al conducir en pistas o carreteras, a veces es difícil mantener los ojos en todas partes a la vez, comprobando el camino por delante, hacia donde girar, dónde disminuir la velocidad o tratar de mantenerla y no acelerar; es por ello, que existen mecanismos destinados a reglamentar el tránsito, advertir o informar a los usuarios mediante palabras, sonidos o símbolos determinados.
- La policía de tránsito o las **señalizaciones vehiculares** regulan el tránsito e informan al usuario sobre direcciones, rutas, advertencias, así como dificultades existentes en las carreteras y previenen cualquier peligro o infracción que podría presentarse durante la circulación vehicular.



Introducción

- Sin embargo, cuando estos mecanismos no son reconocidos o percibidos pueden ocasionar no solo que la congestión del tráfico aumente, sino que también se produzcan accidentes que en muchos casos derivan en consecuencias fatales, generando inseguridad vial.



Introducción

- Sin embargo, cuando estos mecanismos no son reconocidos o percibidos pueden ocasionar no solo que la congestión del tráfico aumente, sino que también se produzcan accidentes que en muchos casos derivan en consecuencias fatales, generando inseguridad vial.
- La inseguridad vial es un problema de interés mundial, según el último informe de la OMS (Organización Mundial de la salud) anualmente cerca de 1,3 millones de personas mueren alrededor del mundo y entre 20 y 50 millones padecen traumatismos no mortales,(OMS, 2017). Son distintas las causas que llevan a este problema, de las cuales las principales pueden ser la falta de concientización y educación vial.



Introducción

- Es por ello que trabajar en obtener vehículos más seguros es un factor fundamental para prevenir de alguna forma los accidentes de tránsito o reducir la probabilidad de que estos sean producidos, (OMS, 2017).



Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

1 Introducción

2 Motivación

3 Formulación del problema e Hipótesis

4 Importancia de la investigación

5 Contribución de la investigación

6 Marco teórico

- Aprendizaje Profundo
- Red Convolutacional
- Arquitectura del Modelo
- Componentes del Modelo
- Entrenamiento y Validación

7 Propuesta o tema central de la tesis

8 Resultados de la tesis

9 Consideraciones finales

10 Referencias bibliograficas



Motivación

- Para contribuir con lo antes mencionado, se han venido planteando formas que permitan la asistencia en el reconocimiento de señales de tránsito, la cual es un problema de clasificación que comúnmente presenta desigualdades en las frecuencias de aparición de las categorías. Además, las señales de tránsito muestran una amplia gama de variaciones entre las clases en términos de color, forma, subconjuntos de clases que son muy similares entre sí y la presencia de símbolos, leyendas o texto. A esto es sumado, las grandes variaciones en las apariencias visuales debido a cambios de iluminación, occlusiones parciales, rotaciones, condiciones meteorológicas, escalamiento, etc.



Motivación

- Para contribuir con lo antes mencionado, se han venido planteando formas que permitan la asistencia en el reconocimiento de señales de tránsito, la cual es un problema de clasificación que comúnmente presenta desigualdades en las frecuencias de aparición de las categorías. Además, las señales de tránsito muestran una amplia gama de variaciones entre las clases en términos de color, forma, subconjuntos de clases que son muy similares entre sí y la presencia de símbolos, leyendas o texto. A esto es sumado, las grandes variaciones en las apariencias visuales debido a cambios de iluminación, occlusiones parciales, rotaciones, condiciones meteorológicas, escalamiento, etc.
- Todo esto representa un reto para el reconocedor/clasificador de señales de tránsito vehicular y es por ello que se han venido realizando diversas investigaciones.



Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

- 1 Introducción
- 2 Motivación
- 3 **Formulación del problema e Hipótesis**
- 4 Importancia de la investigación
- 5 Contribución de la investigación
- 6 Marco teórico
 - Aprendizaje Profundo
 - Red Convolutacional
 - Arquitectura del Modelo
 - Componentes del Modelo
 - Entrenamiento y Validación
- 7 Propuesta o tema central de la tesis
- 8 Resultados de la tesis
- 9 Consideraciones finales
- 10 Referencias bibliograficas



Formulación del problema

En este trabajo, se propone responder a la siguiente pregunta:

¿Cómo se puede reconocer de manera automática una señal de tránsito vehicular?

Hipótesis

Un modelo basado en el aprendizaje profundo de redes neuronales convolucionales permitirá el reconocimiento automático de señales de tránsito vehicular.



Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

- 1 Introducción
- 2 Motivación
- 3 Formulación del problema e Hipótesis
- 4 Importancia de la investigación
- 5 Contribución de la investigación
- 6 Marco teórico
 - Aprendizaje Profundo
 - Red Convolutacional
 - Arquitectura del Modelo
 - Componentes del Modelo
 - Entrenamiento y Validación
- 7 Propuesta o tema central de la tesis
- 8 Resultados de la tesis
- 9 Consideraciones finales
- 10 Referencias bibliograficas



Importancia de la investigación - Justificación Académica

- La importancia de esta investigación en el punto de vista de ciencias de la computación se justifica en poner en práctica los conocimientos adquiridos en la formación académica, siendo los más resaltantes el tema de procesamiento de imágenes e inteligencia artificial, con la finalidad de obtener un modelo robusto de redes neuronales convolucionales basadas en el aprendizaje profundo (deep learning) que permita el reconocimiento de señales de tránsito vehicular.



Importancia de la investigación - Justificación Académica

- La importancia de esta investigación en el punto de vista de ciencias de la computación se justifica en poner en práctica los conocimientos adquiridos en la formación académica, siendo los más resaltantes el tema de procesamiento de imágenes e inteligencia artificial, con la finalidad de obtener un modelo robusto de redes neuronales convolucionales basadas en el aprendizaje profundo (deep learning) que permita el reconocimiento de señales de tránsito vehicular.
- Con los rápidos avances de las estructuras de algoritmos de aprendizaje profundo y la factibilidad de su implementación de alto rendimiento con unidades de procesamiento gráfico (GPU), es ventajoso investigar en problemas de clasificación de imágenes desde la perspectiva de un aprendizaje profundo eficiente, (Gu et al., 2015). Siendo ésta la primera investigación realizada en base a imágenes de señales de tránsito vehicular del Perú.



Importancia de la investigación - Justificación Social

- Teniendo conocimiento de lo descrito sobre la realidad problemática, la visibilidad y conocimiento de señales de tráfico es crucial para la seguridad de los conductores y es por ello que la introducción de un modelo de reconocimiento de señales de tránsito que funcione en diferentes contextos puede formar parte de la solución a que constantes infracciones y en consecuencia evitar o reducir estos índices progresivamente.



Importancia de la investigación - Justificación Social

- Teniendo conocimiento de lo descrito sobre la realidad problemática, la visibilidad y conocimiento de señales de tráfico es crucial para la seguridad de los conductores y es por ello que la introducción de un modelo de reconocimiento de señales de tránsito que funcione en diferentes contextos puede formar parte de la solución a que constantes infracciones y en consecuencia evitar o reducir estos índices progresivamente.
- Algunos ejemplos de aplicaciones inmediatas:



Importancia de la investigación - Justificación Social

- Teniendo conocimiento de lo descrito sobre la realidad problemática, la visibilidad y conocimiento de señales de tráfico es crucial para la seguridad de los conductores y es por ello que la introducción de un modelo de reconocimiento de señales de tránsito que funcione en diferentes contextos puede formar parte de la solución a que constantes infracciones y en consecuencia evitar o reducir estos índices progresivamente.
- Algunos ejemplos de aplicaciones inmediatas:
 - Dar una notificación de ciertas restricciones en el límite de velocidad.



- Teniendo conocimiento de lo descrito sobre la realidad problemática, la visibilidad y conocimiento de señales de tráfico es crucial para la seguridad de los conductores y es por ello que la introducción de un modelo de reconocimiento de señales de tránsito que funcione en diferentes contextos puede formar parte de la solución a que constantes infracciones y en consecuencia evitar o reducir estos índices progresivamente.
- Algunos ejemplos de aplicaciones inmediatas:
 - Dar una notificación de ciertas restricciones en el límite de velocidad.
 - Recibir un aviso de no estacionarse para posteriormente evitar un infracción



Importancia de la investigación - Justificación Social

- Teniendo conocimiento de lo descrito sobre la realidad problemática, la visibilidad y conocimiento de señales de tráfico es crucial para la seguridad de los conductores y es por ello que la introducción de un modelo de reconocimiento de señales de tránsito que funcione en diferentes contextos puede formar parte de la solución a que constantes infracciones y en consecuencia evitar o reducir estos índices progresivamente.
- Algunos ejemplos de aplicaciones inmediatas:
 - Dar una notificación de ciertas restricciones en el límite de velocidad.
 - Recibir un aviso de no estacionarse para posteriormente evitar un infracción
 - Darse cuenta de que está cometiendo una infracción al girar hacia la derecha al recibir un aviso de que se debe girar solo hacia la izquierda



Importancia de la investigación - Justificación Social

- Teniendo conocimiento de lo descrito sobre la realidad problemática, la visibilidad y conocimiento de señales de tráfico es crucial para la seguridad de los conductores y es por ello que la introducción de un modelo de reconocimiento de señales de tránsito que funcione en diferentes contextos puede formar parte de la solución a que constantes infracciones y en consecuencia evitar o reducir estos índices progresivamente.
- Algunos ejemplos de aplicaciones inmediatas:
 - Dar una notificación de ciertas restricciones en el límite de velocidad.
 - Recibir un aviso de no estacionarse para posteriormente evitar un infracción
 - Darse cuenta de que está cometiendo una infracción al girar hacia la derecha al recibir un aviso de que se debe girar solo hacia la izquierda
 - Una aplicación móvil que dé la posibilidad de reconocer automáticamente aquella señal de tránsito que usuarios puedan desconocer serviría como aporte en la educación vial.



Importancia de la investigación - Justificación Social

- Es importante porque a través de un modelo que reconozca señales de tránsito vehicular se podría contribuir en la industria automotriz, específicamente en los sistemas avanzados de asistencia al conductor (del inglés, ADAS); así como también se ha descrito anteriormente, el modelo pretendido puede ser usado para formar parte de diversos mecanismos que buscan dar soluciones a la inseguridad vial.



Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

- 1 Introducción
- 2 Motivación
- 3 Formulación del problema e Hipótesis
- 4 Importancia de la investigación
- 5 Contribución de la investigación
- 6 Marco teórico

- Aprendizaje Profundo
- Red Convolutacional
- Arquitectura del Modelo
- Componentes del Modelo
- Entrenamiento y Validación

- 7 Propuesta o tema central de la tesis
- 8 Resultados de la tesis
- 9 Consideraciones finales
- 10 Referencias bibliograficas



Contribución de la investigación

- Todos los hechos descritos hacen que el reconocimiento de las señales de tránsito sea un reto desafiante y esencial en muchos aspectos, no solo para contribuir en los esfuerzos de la industria automotriz en el campo de la asistencia al conductor, sino también para organismos internacionales y gubernamentales que buscan constantemente introducir nuevos mecanismos y tecnologías que faciliten y mejoren la conducción vehicular. Es por ello que esta investigación contribuye de la siguiente manera:



Contribución de la investigación

- Otorga un modelo computacional para el reconocimiento de 43 tipos de señales de Tránsito de Alemania, con una tasa de acierto del **98.62 %**, mucho mejor que el resultado de 95.29 % obtenido por Ayuque Arenas (2016) y mucho más próximo al mejor resultado de 99.46 % obtenido en las investigaciones hechas por Cireşan et al. (2012) en base al dataset GTSRB.



Contribución de la investigación

- Otorga un modelo computacional para el reconocimiento de 43 tipos de señales de Tránsito de Alemania, con una tasa de acierto del **98.62 %**, mucho mejor que el resultado de 95.29 % obtenido por Ayuque Arenas (2016) y mucho más próximo al mejor resultado de 99.46 % obtenido en las investigaciones hechas por Cireşan et al. (2012) en base al dataset GTSRB.
- Otorga un modelo computacional para el reconocimiento de 7 tipos señales de Tránsito del Perú, el cual posee un (**99.02 %**) de acierto tras analizar una muestra de 4698 imágenes.



Contribución de la investigación

- Otorga un modelo computacional para el reconocimiento de 43 tipos de señales de Tránsito de Alemania, con una tasa de acierto del **98.62 %**, mucho mejor que el resultado de 95.29 % obtenido por Ayuque Arenas (2016) y mucho más próximo al mejor resultado de 99.46 % obtenido en las investigaciones hechas por Cireşan et al. (2012) en base al dataset GTSRB.
- Otorga un modelo computacional para el reconocimiento de 7 tipos señales de Tránsito del Perú, el cual posee un (**99.02 %**) de acierto tras analizar una muestra de 4698 imágenes.
- La investigación ofrece para futuras investigaciones, un dataset de señales de Tránsito del Perú compuesto por 31314 imágenes distribuidas en 7 categorías.



Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

- 1 Introducción
- 2 Motivación
- 3 Formulación del problema e Hipótesis
- 4 Importancia de la investigación
- 5 Contribución de la investigación
- 6 Marco teórico
 - Aprendizaje Profundo
 - Red Convolutacional
 - Arquitectura del Modelo
 - Componentes del Modelo
 - Entrenamiento y Validación
- 7 Propuesta o tema central de la tesis
- 8 Resultados de la tesis
- 9 Consideraciones finales
- 10 Referencias bibliograficas



Aprendizaje Profundo

- Aprendizaje Profundo



Aprendizaje Profundo

● Aprendizaje Profundo

- ▶ El verdadero desafío para la inteligencia artificial fue y es resolver las tareas que son fáciles de realizar para las personas pero difíciles de describir de manera formal, problemas que resolvemos intuitivamente, que se sienten automáticos, como reconocer palabras, rostros u objetos en las imágenes (Goodfellow et al., 2016).



Aprendizaje Profundo

● Aprendizaje Profundo

- ▶ El verdadero desafío para la inteligencia artificial fue y es resolver las tareas que son fáciles de realizar para las personas pero difíciles de describir de manera formal, problemas que resolvemos intuitivamente, que se sienten automáticos, como reconocer palabras, rostros u objetos en las imágenes (Goodfellow et al., 2016).
- ▶ El aprendizaje automático es considerada como la mejor técnica de la inteligencia artificial(IA); es decir, es el campo de la IA que hoy en día muestra la mayor promesa al proporcionar herramientas que la industria y la sociedad pueden usar para producir algún cambio, (Goodfellow et al., 2016).



Aprendizaje Profundo

● Aprendizaje Profundo

- ▶ El verdadero desafío para la inteligencia artificial fue y es resolver las tareas que son fáciles de realizar para las personas pero difíciles de describir de manera formal, problemas que resolvemos intuitivamente, que se sienten automáticos, como reconocer palabras, rostros u objetos en las imágenes (Goodfellow et al., 2016).
- ▶ El aprendizaje automático es considerada como la mejor técnica de la inteligencia artificial(IA); es decir, es el campo de la IA que hoy en día muestra la mayor promesa al proporcionar herramientas que la industria y la sociedad pueden usar para producir algún cambio, (Goodfellow et al., 2016).
- ▶ En tal sentido,el aprendizaje automático toma algunas de las ideas centrales de la inteligencia artificial y las enfoca en resolver problemas del mundo real con redes neuronales diseñadas para imitar nuestra propia toma de decisiones.



Aprendizaje Profundo

- Por otro lado, el aprendizaje profundo o avanzado de máquinas (del inglés, **deep learning**) se centra aún más estrechamente en un subconjunto de herramientas y técnicas del aprendizaje automático y los aplica a la solución de casi cualquier problema que requiera "pensamiento", ya sea humano o artificial (Goodfellow et al., 2016).



Aprendizaje Profundo

- Por otro lado, el aprendizaje profundo o avanzado de máquinas (del inglés, **deep learning**) se centra aún más estrechamente en un subconjunto de herramientas y técnicas del aprendizaje automático y los aplica a la solución de casi cualquier problema que requiera "pensamiento", ya sea humano o artificial (Goodfellow et al., 2016).
- Se puede sostener que el aprendizaje automático es el único enfoque viable para construir sistemas de inteligencia artificial que puedan operar en entornos complicados del mundo real y el **aprendizaje profundo** es, a su vez, un tipo particular de aprendizaje automático que logra gran poder y flexibilidad al representar el mundo como una jerarquía de conceptos anidados, con cada concepto definido en relación con conceptos más simples y representaciones más abstractas calculadas en base a representaciones menos abstractas, (Goodfellow et al., 2016).



Aprendizaje Profundo

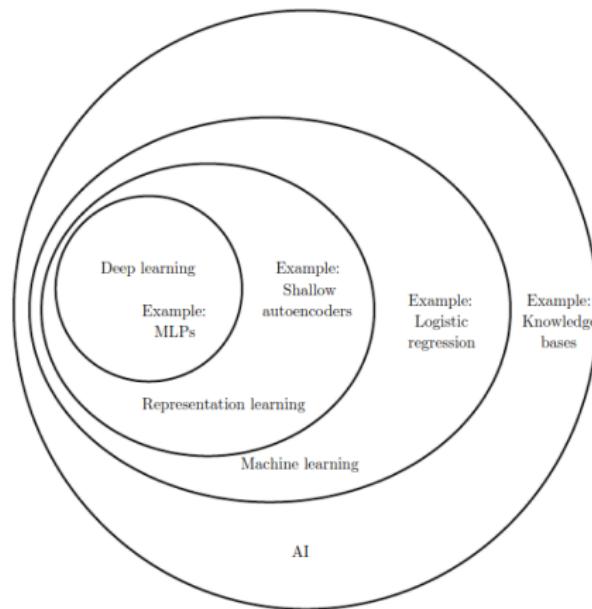


Figura: Diagrama de Venn donde cada sección incluye un ejemplo de una tecnología de IA (ilustra la relación entre estas diferentes disciplinas de la IA)



Fuente: Goodfellow et al. (2016)

Aprendizaje Profundo

- El aprendizaje profundo es un enfoque específico utilizado para construir y entrenar redes neuronales para la toma de decisiones altamente prometedoras, (Goodfellow et al., 2016).



Aprendizaje Profundo

- El aprendizaje profundo es un enfoque específico utilizado para construir y entrenar redes neuronales para la toma de decisiones altamente prometedoras, (Goodfellow et al., 2016).
- Se considera que un algoritmo es profundo si los datos de entrada se pasan a través de una serie de no linealidades o transformaciones no lineales antes de que se emita.



Aprendizaje Profundo

- El aprendizaje profundo es un enfoque específico utilizado para construir y entrenar redes neuronales para la toma de decisiones altamente prometedoras, (Goodfellow et al., 2016).
- Se considera que un algoritmo es profundo si los datos de entrada se pasan a través de una serie de no linealidades o transformaciones no lineales antes de que se emita.
- Este enfoque permite que las computadoras aprendan de la experiencia y entiendan el mundo en términos de una jerarquía de conceptos, con cada concepto definido a través de su relación con conceptos más simples.



Aprendizaje Profundo

- El aprendizaje profundo es un enfoque específico utilizado para construir y entrenar redes neuronales para la toma de decisiones altamente prometedoras, (Goodfellow et al., 2016).
- Se considera que un algoritmo es profundo si los datos de entrada se pasan a través de una serie de no linealidades o transformaciones no lineales antes de que se emita.
- Este enfoque permite que las computadoras aprendan de la experiencia y entiendan el mundo en términos de una jerarquía de conceptos, con cada concepto definido a través de su relación con conceptos más simples.
- Al reunir el conocimiento de la experiencia, este enfoque evita la necesidad de que los operadores humanos especifiquen formalmente todo el conocimiento que necesita la computadora.



Aprendizaje Profundo

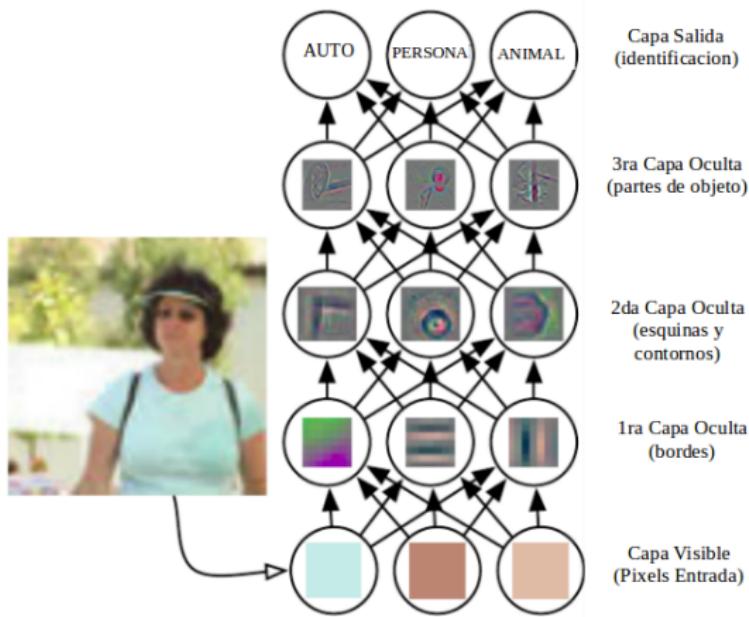


Figura: El aprendizaje profundo permite a la computadora construir conceptos complejos a partir de conceptos más simples



Fuente: Elaboración propia

Red Convolucional

- Nueve de cada diez veces, cuando se escucha que el aprendizaje profundo rompe una nueva barrera tecnológica, las redes neuronales convolucionales están involucradas. También llamados CNN(del inglés , **Convolutional Neural Networks**) o **ConvNets**, estas son las preferidas del campo de redes neuronales profundas. Han aprendido a clasificar las imágenes en categorías incluso mejor que los humanos en algunos casos, (Rohrer, 2016).



Red Convolucional

- Nueve de cada diez veces, cuando se escucha que el aprendizaje profundo rompe una nueva barrera tecnológica, las redes neuronales convolucionales están involucradas. También llamados CNN(del inglés , **Convolutional Neural Networks**) o **ConvNets**, estas son las preferidas del campo de redes neuronales profundas. Han aprendido a clasificar las imágenes en categorías incluso mejor que los humanos en algunos casos, (Rohrer, 2016).
- Estas son muy similares a las redes neuronales ordinarias, están formadas por neuronas que tienen pesos y biases(sesgos) que se pueden aprender información durante el entrenamiento de estas. Cada neurona recibe algunas entradas y realiza operaciones matemáticas.



Red Convolucional

- Una ConvNet se caracteriza por tener una secuencia de capas donde cada una de estas transforma un volumen de activaciones en otro nuevo a través de funciones y el aprendizaje profundo se produce cuando se utilizan varias de estas capas variando los parámetros de configuración dentro y entre dichas capas.



Red Convolucional

- Una ConvNet se caracteriza por tener una secuencia de capas donde cada una de estas transforma un volumen de activaciones en otro nuevo a través de funciones y el aprendizaje profundo se produce cuando se utilizan varias de estas capas variando los parámetros de configuración dentro y entre dichas capas.
- El resultado de las CNNs es que pueden encontrar si una característica está en una imagen sin preocuparse exactamente de donde está. Esto ayuda a resolver el problema de las computadoras al comparar imágenes de manera hiper-literal, es decir, que coincida pixel a pixel para que se trate de imágenes iguales.



Red Convolucional

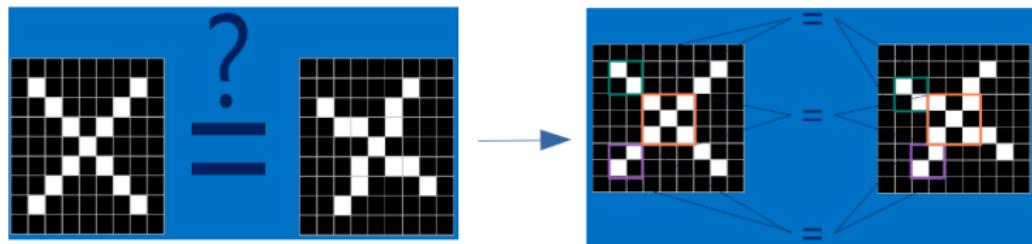


Figura: Análisis de una CNN
Fuente: Rohrer (2016)



Red Convolucional

- Red Convolucional - Terminologías



Red Convolucional

- **Red Convolucional - Terminologías**

- ▶ Una es cuando la red convolucional es vista como un número largo de capas simples y cada paso del procesamiento se considera como una capa en sí misma.



Red Convolucional

● Red Convolucional - Terminologías

- ▶ Una es cuando la red convolucional es vista como un número largo de capas simples y cada paso del procesamiento se considera como una capa en sí misma.
- ▶ Otra terminología es cuando la red convolucional es vista como un número pequeño de capas relativamente complejas, donde cada capa tiene múltiples etapas.



Red Convolucional

● Red Convolucional - Terminologías

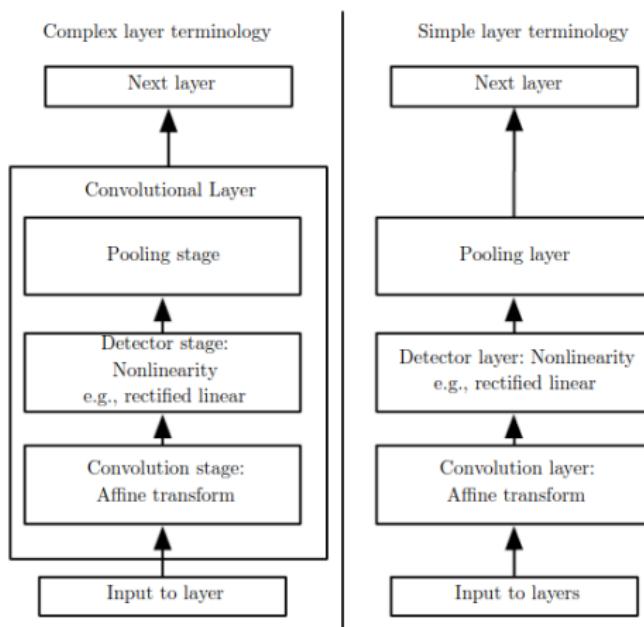


Figura: Terminología de capas complejas(izquierda) y de capas simples(derecha)

Fuente: Goodfellow et al. (2016)



Arquitectura del Modelo

- La arquitectura de la red está inspirada en la arquitectura AlexNet(Krizhevsky et al., 2012) y en la arquitectura Inception (Szegedy et al., 2014) para la clasificación de imágenes.



Arquitectura del Modelo

- La arquitectura de la red está inspirada en la arquitectura AlexNet(Krizhevsky et al., 2012) y en la arquitectura Inception (Szegedy et al., 2014) para la clasificación de imágenes.
- La **arquitectura AlexNet** hizo que las redes convolucionales fueran populares cuando en el concurso ImageNet LSVRC-2012 superó significativamente a todas las otras redes neuronales a través de su diseño de red convolucional profunda para clasificar 1.2 millones de imágenes en 1000 clases diferentes. Ganó el desafío al obtener un error de 15.3 % y el segundo lugar (que no fue una variación de una CNN) obtuvo una tasa de error alrededor del 26,2 % (Krizhevsky et al., 2012).



Arquitectura del Modelo

- Constituida de capas convolucionales y capas totalmente conectadas, donde la salida de cada capa convolucional y cada capa completamente conectada se activa a través de la función no linear RELU. Presenta filtros de tamaño 11x11, 5x5 , 3x3 y capas de agrupación(pooling), siendo lo más destacable el uso de dropout como técnica de regularización y el uso del proceso de Data Augmentation para conseguir esos excelentes resultados.



Arquitectura del Modelo

- Constituida de capas convolucionales y capas totalmente conectadas, donde la salida de cada capa convolucional y cada capa completamente conectada se activa a través de la función no linear RELU. Presenta filtros de tamaño 11x11, 5x5 , 3x3 y capas de agrupación(pooling), siendo lo más destacable el uso de dropout como técnica de regularización y el uso del proceso de Data Augmentation para conseguir esos excelentes resultados.
- La **Arquitectura Inception** es una arquitectura de red neuronal convolucional profunda, creada por un grupo de investigación de Google que fue responsable de establecer el nuevo estado del arte de la técnica para la clasificación y detección en la competencia de reconocimiento visual a gran escala ImageNet 2014(ILSVRC14).



Arquitectura del Modelo

- El principal sello distintivo de esta arquitectura es la utilización mejorada de los recursos informáticos dentro de la red. Esto fue logrado gracias al aumento de la profundidad y el ancho de la red mientras se mantiene el costo computacional constante, reduciendo el número de parámetros usados en AlexNet(60 millones) a 4 millones,(Szegedy et al., 2014).



Arquitectura del Modelo

- El principal sello distintivo de esta arquitectura es la utilización mejorada de los recursos informáticos dentro de la red. Esto fue logrado gracias al aumento de la profundidad y el ancho de la red mientras se mantiene el costo computacional constante, reduciendo el número de parámetros usados en AlexNet(60 millones) a 4 millones,(Szegedy et al., 2014).
- Para optimizar la calidad, las decisiones para elaborar la arquitectura se basaron en el principio Hebbiano y la intuición de funciones procesamiento a escala múltiple, (Szegedy et al., 2014), lo que significa que la salida de las capas convolucionales no solo se envía a la capa posterior, sino que también se ramifica y se introduce al clasificador (capa totalmente conectada).



Arquitectura del Modelo

- La razón detrás de esto es que cuando el clasificador está tomando una decisión basada en convoluciones, podría encontrar que la salida de la primera o segunda capa convolucional también es útil. Básicamente con las características de escala múltiple depende del clasificador qué nivel de abstracción usar, ya que tiene acceso a las salidas de todas las capas convolucionales, es decir, características en todos los niveles de abstracción.



Arquitectura del Modelo

- La razón detrás de esto es que cuando el clasificador está tomando una decisión basada en convoluciones, podría encontrar que la salida de la primera o segunda capa convolucional también es útil. Básicamente con las características de escala múltiple depende del clasificador qué nivel de abstracción usar, ya que tiene acceso a las salidas de todas las capas convolucionales, es decir, características en todos los niveles de abstracción.
- El modelo creado apartir de esta arquitectura llamado GoogLeNet, una red de 22 capas de profundidad, cuya calidad se evalúa en el contexto de reconocimiento y detección, logró tener una tasa de error del 6.67 %, muy cercano al rendimiento a nivel humano.



Arquitectura del Modelo

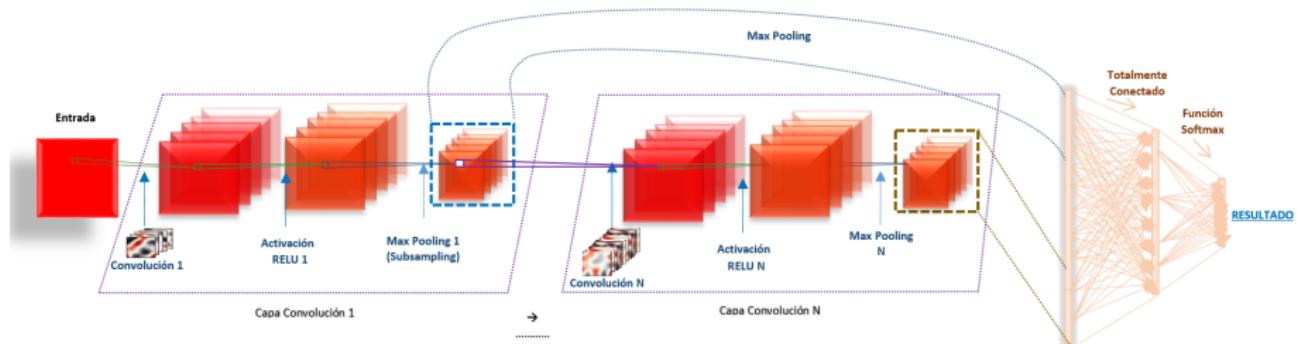


Figura: Modelo del diseño de la Red propuesta

Fuente propia



Componentes del Modelo

- Capa Convolucional



Componentes del Modelo

- Capa Convolucional
- Capa de Activación ReLU(Rectified Linear Units)



Componentes del Modelo

- Capa Convolucional
- Capa de Activación ReLU(Rectified Linear Units)
- Capa de Agrupación(Pooling)



Componentes del Modelo

- Capa Convolucional
- Capa de Activación ReLU(Rectified Linear Units)
- Capa de Agrupación(Pooling)
- Capa totalmente conectada (Fully-connected layer)



Capa Convolucional

- Los parámetros de la capa convolucional consisten básicamente en dos datos. La entrada y todo lo que respecta a un conjunto de filtros(también denominados kernels) cuyos valores se aprenden, es decir, empiezan con datos aleatorios y conforme avance el entrenamiento se van alterando.



Capa Convolucional

- Los parámetros de la capa convolucional consisten básicamente en dos datos. La entrada y todo lo que respecta a un conjunto de filtros(también denominados kernels) cuyos valores se aprenden, es decir, empiezan con datos aleatorios y conforme avance el entrenamiento se van alterando.
- Durante el proceso hacia adelante, se desliza (más precisamente, convolve) cada filtro a través del ancho y alto del volumen de entrada para calcular los productos de puntos entre las entradas del filtro y la entrada en cualquier posición.



Capa Convolucional

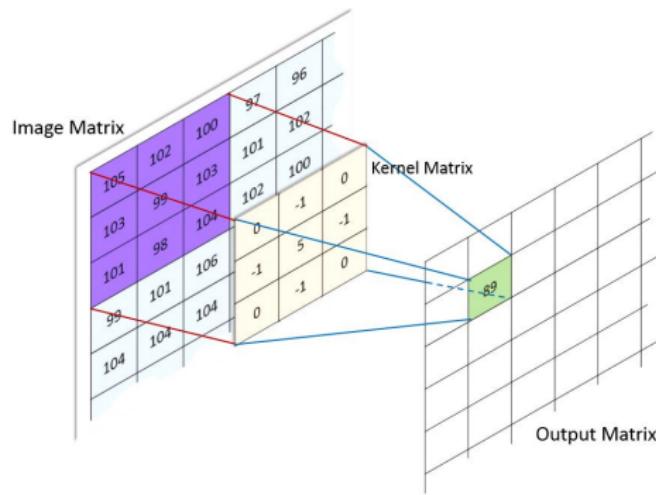


Figura: Posicionamiento del kernel/filtro por pixel

Fuente: Fuente Propia



Capa Convolucional

- A medida que deslizamos el filtro sobre el ancho y la altura del volumen de entrada produciremos un mapa de activación bidimensional que proporciona las respuestas de ese filtro en cada posición espacial. Por lo que simplemente se multiplica cada píxel en el filtro por el valor del píxel en la imagen. Para luego, sumar las respuestas y dividirlas por el número total de píxeles en el filtro.

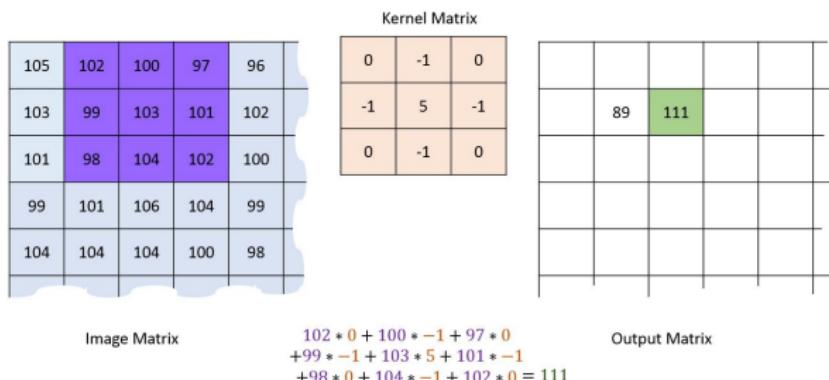


Figura: Cálculo Convolucional



Fuente: Fuente Propia

Capa Convolucional



Capa Convolucional

- Intuitivamente, la red aprenderá los filtros que se activan cuando ven algún tipo de característica visual, como un borde o contorno en alguna orientación específica. Debido a que la convolución aprovecha tres ideas importantes que pueden ayudar a mejorar un sistema de aprendizaje automático:



Capa Convolucional

- Intuitivamente, la red aprenderá los filtros que se activan cuando ven algún tipo de característica visual, como un borde o contorno en alguna orientación específica. Debido a que la convolución aprovecha tres ideas importantes que pueden ayudar a mejorar un sistema de aprendizaje automático:
 - ▶ Interacciones dispersas



Capa Convolucional

- Intuitivamente, la red aprenderá los filtros que se activan cuando ven algún tipo de característica visual, como un borde o contorno en alguna orientación específica. Debido a que la convolución aprovecha tres ideas importantes que pueden ayudar a mejorar un sistema de aprendizaje automático:
 - ▶ Interacciones dispersas
 - ▶ Uso compartido de parámetros



Capa Convolucional

- Intuitivamente, la red aprenderá los filtros que se activan cuando ven algún tipo de característica visual, como un borde o contorno en alguna orientación específica. Debido a que la convolución aprovecha tres ideas importantes que pueden ayudar a mejorar un sistema de aprendizaje automático:
 - ▶ Interacciones dispersas
 - ▶ Uso compartido de parámetros
 - ▶ Representaciones equivalentes



Capa Convolucional

- Capa Convolucional - Interacciones dispersas

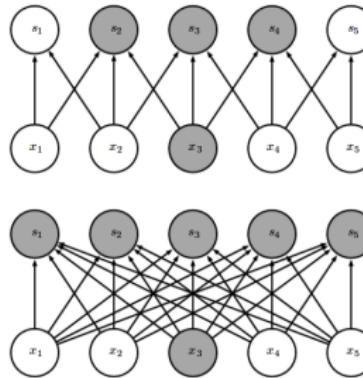


Figura: Conectividad dispersa vs No dispersa
Fuente:(Rohrer, 2016)

Cuando \mathbf{s} está formado por convolución con un kernel de ancho 3, solo tres salidas se ven afectadas por \mathbf{x} . (Abajo) Cuando \mathbf{s} está formado por la multiplicación de la matriz, la conectividad ya no es dispersa, por lo que todos los resultados se ven afectados por x_3 .



Capa Convolucional

- Capa Convolucional - Interacciones dispersas



Capa Convolucional

- **Capa Convolucional - Interacciones dispersas**

- ▶ Esto significa que necesitamos almacenar menos parámetros, lo que reduce los requisitos de memoria del modelo y mejora su eficiencia estadística. También significa que el cálculo de la salida requiere menos operaciones, (Goodfellow et al., 2016).



Capa Convolucional

- Capa Convolucional - Uso compartido de parámetros



Capa Convolucional

- **Capa Convolucional - Uso compartido de parámetros**

- ▶ En una red neuronal convolucional, cada miembro del kernel se utiliza en cada posición de la entrada (excepto tal vez algunos de los píxeles de los bordes, dependiendo de las decisiones de diseño con respecto al límite). El uso compartido de parámetros utilizado por la operación de convolución significa que en lugar de aprender conjuntos de parámetros para cada ubicación por separado, aprendemos un solo conjunto por kernel, (Goodfellow et al., 2016).



Capa Convolucional

- Capa Convolucional - Representaciones equivalentes



Capa Convolucional

- **Capa Convolucional - Representaciones equivalentes**

- ▶ La forma particular de compartir los parámetros hace que la capa tenga una propiedad llamada **representaciones equivalentes**. Decir que una función es equivalente significa que si la entrada cambia, la salida cambia de la misma manera.



Capa Convolucional

- **Capa Convolucional - Representaciones equivalentes**

- ▶ La forma particular de compartir los parámetros hace que la capa tenga una propiedad llamada **representaciones equivalentes**. Decir que una función es equivalente significa que si la entrada cambia, la salida cambia de la misma manera.
- ▶ La convolución crea un mapa en 2-D de donde aparecen ciertas características en la entrada. Si movemos el objeto en la entrada, su representación se moverá la misma cantidad en la salida.



Capa Convolucional

- Capa Convolucional - Representaciones equivalentes

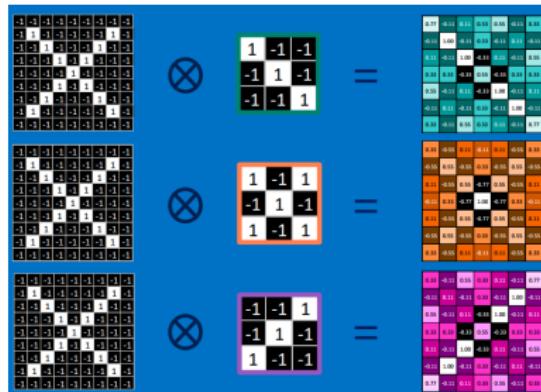


Figura: Resultado de Convolución(conjunto de mapas de activación. generado a partir de 3 filtros para 3 características: diagonal derecha, cruzamiento central y diagonal izquierda). Simbolo de convolución: \otimes

Fuente: Rohrer (2016)



Capa Convolucional

• Capa Convolucional - Construcción de un filtro o kernel

- 1 La extensión espacial es el tamaño del filtro, comúnmente es de tamaño impar tanto en largo y ancho.
- 2 El stride es otra pieza del bloque de construcción básico de los filtros convolucionales. Este representa el 'paso' en la operación de convolución indicando cuánto es que se debe desplazar un filtro en una imagen con cada paso. El filtro se desliza sobre la imagen, se detiene en cada longitud de salto y realiza las operaciones necesarias en ese paso.

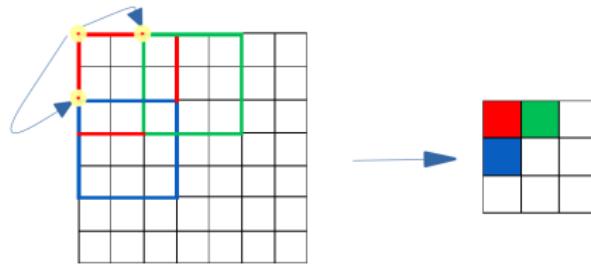


Figura: Imagen con stride igual a 2, para el filtro tanto en largura como anchura

Fuente propia



Capa Convolucional

• Capa Convolucional - Construcción de un filtro o kernel

- 3 Zero-padding agrega ceros alrededor del borde de una imagen.

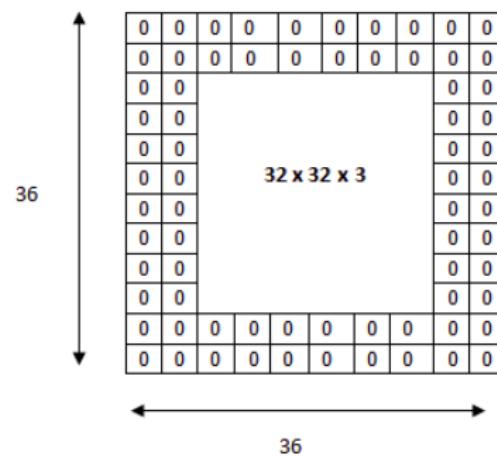


Figura: Ejemplo de zero-padding con tamaño 2

Fuente propia



Capa Convolucional

- Capa Convolucional - Ecuación de Convolución



Capa Convolucional

- **Capa Convolucional - Ecuación de Convolución**

- $$\text{conv}_j^n = \sum_{k=1}^k x_k^n \times w_{kj}^n + b_n$$

En el que:

- ▶ x son valores de entrada
- ▶ w, b pesos y biases(sesgos) del kernel, respectivamente
- ▶ n es el número de la capa
- ▶ j es el número del filtro de salida
- ▶ k es la cantidad de filtros en la capa $n - 1$ o n



Capa Convolucional

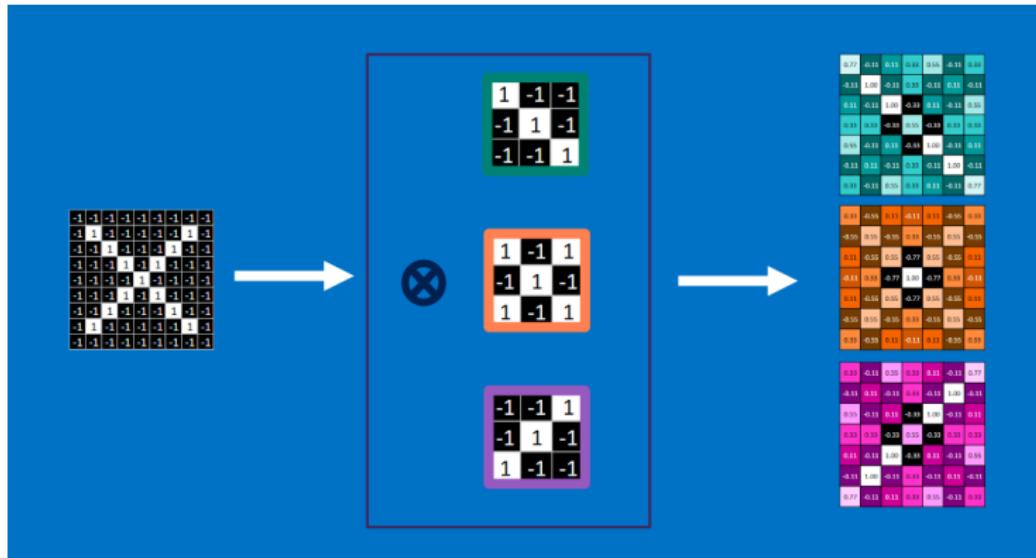


Figura: Resultado Final de Convolución

Fuente: Rohrer (2016)



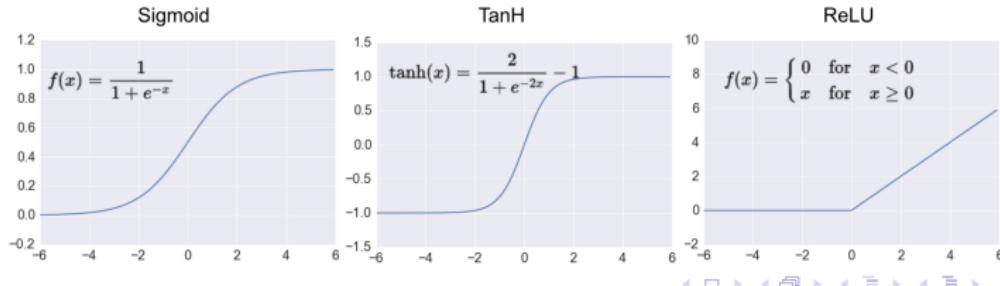
Capa ReLU(Rectified Linear Units)

- Una red neuronal artificial consiste básicamente en multiplicaciones y suma de matrices. Si solo utilizáramos estos cálculos lineales, podríamos apilarlos uno encima del otro y esa no sería una red muy profunda. Por lo tanto, a menudo se utiliza funciones de activación no lineales en cada capa de la red.



Capa ReLU(Rectified Linear Units)

- Una red neuronal artificial consiste básicamente en multiplicaciones y suma de matrices. Si solo utilizáramos estos cálculos lineales, podríamos apilarlos uno encima del otro y esa no sería una red muy profunda. Por lo tanto, a menudo se utiliza funciones de activación no lineales en cada capa de la red.
- Estas son las tres funciones de activación no lineal más populares:
 - 1) Sigmoid (analiza un valor entre 0 y 1)
 - 2) TanH (analiza un valor entre -1 y 1)
 - 3) ReLU (si el valor es negativo, se convierte en 0, de lo contrario, permanece igual)



Capa ReLU(Rectified Linear Units)

- Actualmente, ReLU es la función de activación no lineal más utilizada (Karpathy, 2016), que básicamente aplica la función $f(x) = \max(0, x)$ a todos los valores en el volumen de entrada.



Capa ReLU(Rectified Linear Units)

- Actualmente, ReLU es la función de activación no lineal más utilizada (Karpathy, 2016), que básicamente aplica la función $f(x) = \max(0, x)$ a todos los valores en el volumen de entrada.
 - La red puede entrenar mucho más rápido (debido a la eficiencia computacional) sin hacer una diferencia significativa en la precisión.



Capa ReLU(Rectified Linear Units)

- Actualmente, ReLU es la función de activación no lineal más utilizada (Karpathy, 2016), que básicamente aplica la función $f(x) = \max(0, x)$ a todos los valores en el volumen de entrada.
 - La red puede entrenar mucho más rápido (debido a la eficiencia computacional) sin hacer una diferencia significativa en la precisión.
 - Alivia el problema del gradiente de fuga, que es el problema donde las capas inferiores de la red entran muy lentamente porque el gradiente de optimización disminuye exponencialmente a través de las capas.



Capa ReLU(Rectified Linear Units)

- Actualmente, ReLU es la función de activación no lineal más utilizada (Karpathy, 2016), que básicamente aplica la función $f(x) = \max(0, x)$ a todos los valores en el volumen de entrada.
 - ▶ La red puede entrenar mucho más rápido (debido a la eficiencia computacional) sin hacer una diferencia significativa en la precisión.
 - ▶ Alivia el problema del gradiente de fuga, que es el problema donde las capas inferiores de la red entran muy lentamente porque el gradiente de optimización disminuye exponencialmente a través de las capas.
- El hecho de que entradas en la función de activación de valores menores o iguales a cero resulten cero, induce a la dispersión en las unidades ocultas, que según lo comentado anteriormente, produce representaciones dispersas las cuales se consideran más valiosas, (Nair and Hinton, 2010).



Capa ReLU(Rectified Linear Units)

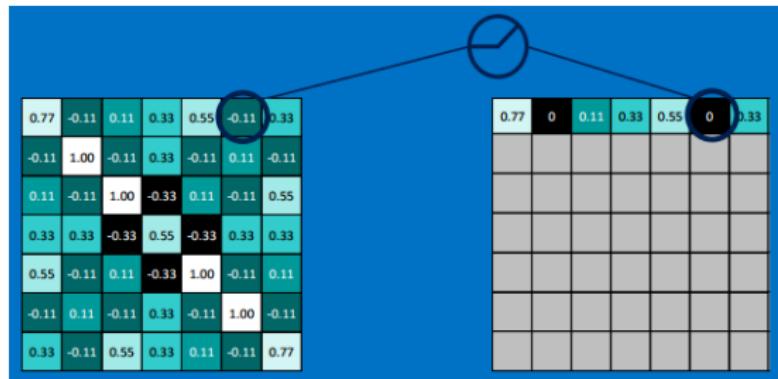


Figura: Procedimiento de la función ReLU

Fuente: Rohrer (2016)



Capa ReLU(Rectified Linear Units)

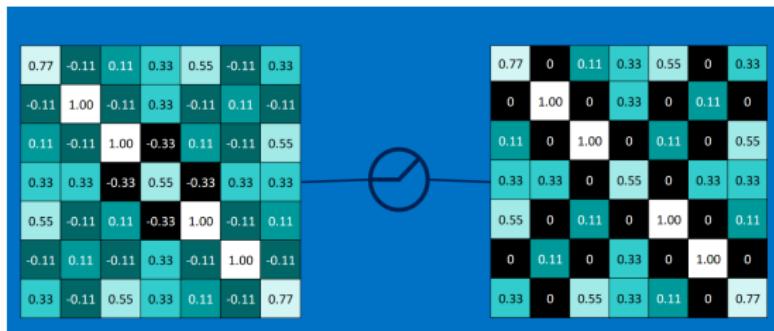


Figura: Procedimiento de la función ReLU

Fuente: Rohrer (2016)



Capa ReLU(Rectified Linear Units)



Figura: Procedimiento de la función ReLU

Fuente: Rohrer (2016)



Técnica Dropout

- Es una técnica de regularización que tiene por objetivo de reducir el sobreajuste que puede darse durante el entrenamiento de una red neuronal. Consiste en establecer a cero la salida de cada neurona oculta con una probabilidad 0.5(comúnmente). Las neuronas que se abandonan de esta manera no contribuyen al pase directo y no participan en las siguientes etapas de entrenamiento.

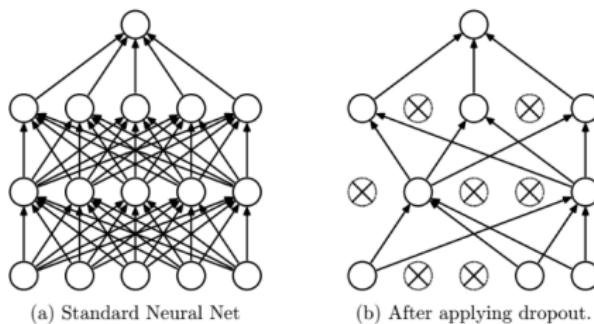


Figura: En la izquierda la red neuronal común y a la derecha la red neuronal diluida producida por la aplicación de dropout

(Romero, 2015a)



Técnica Dropout

- Por lo tanto, cada vez que se presenta una entrada, la red neuronal muestra una arquitectura diferente, pero todas estas arquitecturas comparten ponderaciones. Esta técnica reduce las coadaptaciones complejas de las neuronas, ya que una neurona no puede depender de la presencia de otras neuronas particulares. Por lo tanto, se ve forzado a aprender características más robustas que son útiles en conjunción con muchos subconjuntos aleatorios diferentes de las otras neuronas.

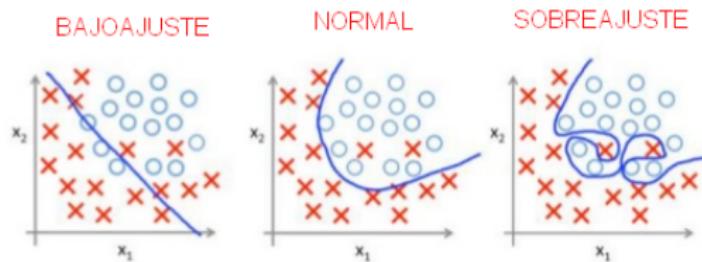


Figura: Procesos que pueden ocurrir durante entrenamiento. Dropout evita el sobreajuste

Fuente propia



Capa de Agrupación(Pooling)

- Una variedad de cálculos que reducen la dimensionalidad de un mapa de características, permitiendo que la red sea robusta e invariante a pequeños cambios y distorsiones.



Capa de Agrupación(Pooling)

- Una variedad de cálculos que reducen la dimensionalidad de un mapa de características, permitiendo que la red sea robusta e invariante a pequeños cambios y distorsiones.
- La capa Pooling(también conocida como una capa de reducción de resolución) combina o agrupa, un conjunto de valores en su campo receptivo en un menor número de valores.



Capa de Agrupación(Pooling)

- Una variedad de cálculos que reducen la dimensionalidad de un mapa de características, permitiendo que la red sea robusta e invariante a pequeños cambios y distorsiones.
- La capa Pooling(también conocida como una capa de reducción de resolución) combina o agrupa, un conjunto de valores en su campo receptivo en un menor número de valores.
- Puede ser configurado en función del tamaño de su campo receptivo (por ejemplo, 2×2) y en función a la operación de agrupamiento (por ejemplo, máximo-max o promedio-average).

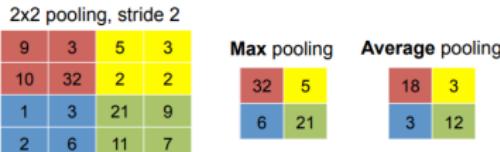


Figura: Formas de agrupamiento(pooling)

Fuente: Jia et al. (2014)



Capa de Agrupación(Pooling)

- El Max-pooling es la más popular reduciendo progresivamente el tamaño espacial de la representación de una imagen(mapa de características) mientras conserva la información más importante en ella. El proceso matematico consiste en pasar una pequeña ventana(kernel)através de una imagen y tomar el valor máximo de la ventana en cada paso.



Figura: Operación MAX-pooling en capa de Agrupación

Fuente: Rohrer (2016)



Capa de Agrupación(Pooling)

- Este proceso es aplicado para cada mapa de activación(salida de la capa de convolución). Análogamente con la capa de convolución, el resultado en esta capa es un conjunto de imágenes que muestran una versión agrupada de las imágenes de entrada.



Figura: Resultado de Agrupación



Componentes del Modelo - Capas Conectadas

- Finalmente la salida de cada capa se convierte en entrada para una posterior.

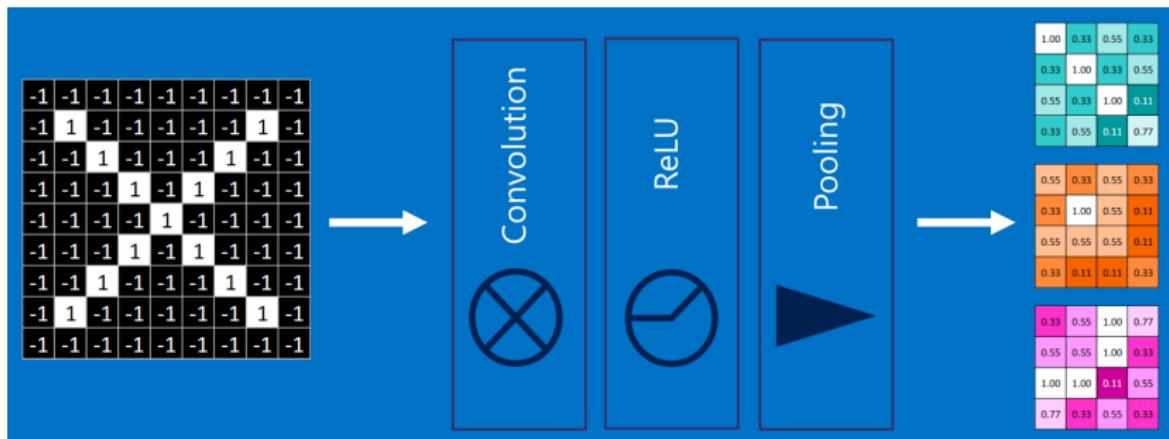


Figura: Capas Apiladas

Fuente: Rohrer (2016)



Componentes del Modelo - Capas Conectadas

- Las capas pueden ser repetidas múltiples veces.

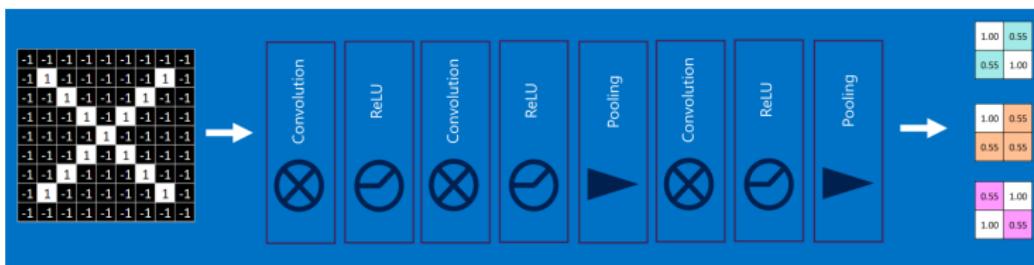


Figura: Apilación Profunda de Capas

Fuente: Rohrer (2016)



Capa totalmente conectada (Fully-connected layer)

- Esta capa por lo general aparece al final de la arquitectura y es similar al Perceptrón multicapa (Multilayer Perceptron -MLP), en el cual la neurona de salida se conecta a todas las neuronas de entrada y el peso de las conexiones son actualizadas usando el método de retropropagación.

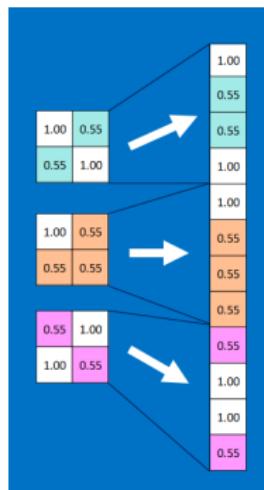


Figura: Eventualmente, con un mapa de características lo suficientemente pequeño, el contenido se aplastará en un vector de una sola dimensión y será entrada para en un MLP totalmente conectado para su procesamiento



Fuente: Rohrer (2016)

Capa totalmente conectada (Fully-connected layer)

- En esta última capa del modelo generalmente se utiliza una función de activación diferente, porque en esta capa se pretende que tenga una salida determinada que represente la probabilidad de predicción del clasificador. La **función softmax**, conocida también como función exponencial normalizada, es muy popular cuando se hace la clasificación.



Capa totalmente conectada (Fully-connected layer)

- En esta última capa del modelo generalmente se utiliza una función de activación diferente, porque en esta capa se pretende que tenga una salida determinada que represente la probabilidad de predicción del clasificador. La **función softmax**, conocida también como función exponencial normalizada, es muy popular cuando se hace la clasificación.
- Esta función permitirá estimar la probabilidad de que la imagen de entrada pertenezca a cada una de las clases al realizar una normalización con el objetivo que el valor de cada neurona esté limitado entre cero y uno y así permitir que el resultado de las neuronas de salida sumen a uno, (Bishop, 2006).



Entrenamiento de Red

- La red procesa los registros en los datos de entrenamiento usando los pesos, biases y las funciones en las capas ocultas, luego compara las salidas resultantes con las salidas deseadas. Los errores se propagan a través del sistema, lo que hace que el sistema ajuste los pesos y biases que serán procesados en la siguiente iteración de entrenamiento. Este proceso ocurre una y otra vez sobre un conjunto de datos, a medida que los pesos se ajustan (refinan) continuamente.



Entrenamiento de Red

- La red procesa los registros en los datos de entrenamiento usando los pesos, biases y las funciones en las capas ocultas, luego compara las salidas resultantes con las salidas deseadas. Los errores se propagan a través del sistema, lo que hace que el sistema ajuste los pesos y biases que serán procesados en la siguiente iteración de entrenamiento. Este proceso ocurre una y otra vez sobre un conjunto de datos, a medida que los pesos se ajustan (refinan) continuamente.
- Esto se repite con cada imagen subsiguiente en el conjunto de imágenes etiquetadas. Los problemas que ocurren en una sola imagen se olvidan rápidamente, pero los patrones que aparecen en muchas imágenes se integran en las características(pesos y biases de conexión).



Entrenamiento de Red

- La red procesa los registros en los datos de entrenamiento usando los pesos, biases y las funciones en las capas ocultas, luego compara las salidas resultantes con las salidas deseadas. Los errores se propagan a través del sistema, lo que hace que el sistema ajuste los pesos y biases que serán procesados en la siguiente iteración de entrenamiento. Este proceso ocurre una y otra vez sobre un conjunto de datos, a medida que los pesos se ajustan (refinan) continuamente.
- Esto se repite con cada imagen subsiguiente en el conjunto de imágenes etiquetadas. Los problemas que ocurren en una sola imagen se olvidan rápidamente, pero los patrones que aparecen en muchas imágenes se integran en las características(pesos y biases de conexión).
- Para realizar este proceso, existe un método de optimización muy popular denominado **Descenso de gradiente**.



Entrenamiento de Red - Descenso de Gradiente

- Un gradiente mide cuánto cambia la salida de una función si se cambia un poco las entradas.



Entrenamiento de Red - Descenso de Gradiente

- Un gradiente mide cuánto cambia la salida de una función si se cambia un poco las entradas.
- En el caso del entrenamiento de una red neuronal, el gradiente simplemente mide el cambio en todos los pesos con respecto al cambio en el error. El gradiente se puede representar como la pendiente de una función, cuanto más alto es el gradiente, más pronunciada es la pendiente y más rápido puede aprender un modelo, (Dong and Zhou, 2008).



Entrenamiento de Red - Descenso de Gradiente

- Un gradiente mide cuánto cambia la salida de una función si se cambia un poco las entradas.
- En el caso del entrenamiento de una red neuronal, el gradiente simplemente mide el cambio en todos los pesos con respecto al cambio en el error. El gradiente se puede representar como la pendiente de una función, cuanto más alto es el gradiente, más pronunciada es la pendiente y más rápido puede aprender un modelo, (Dong and Zhou, 2008).
- La idea detrás del descenso de gradiente es disminuir de forma gradual, pero constante, el error de salida ajustando los pesos y biases.

$$\frac{\partial E}{\partial w_{ij}}$$

Ecuación de la derivada del error con respecto al peso



Entrenamiento de Red - Descenso de Gradiente

- Una vez que encontramos esta derivada, actualizamos el peso a través de la siguiente fórmula:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$$

Representación del ajuste del peso, donde η es la tasa de aprendizaje



Entrenamiento de Red - Descenso de Gradiente

- Una vez que encontremos esta derivada, actualizamos el peso a través de la siguiente fórmula:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$$

Representación del ajuste del peso, donde η es la tasa de aprendizaje

- La **tasa de aprendizaje**(Learning Rate) puede entenderse como el tamaño del paso en la optimización para minimizar la función de pérdida de la red, es decir, es un parámetro que determina cuánto influye un paso de actualización en el valor actual de los pesos, (Jordan, 2018).



Entrenamiento de Red - Descenso de Gradiente

- Una vez que encontramos esta derivada, actualizamos el peso a través de la siguiente fórmula:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$$

Representación del ajuste del peso, donde η es la tasa de aprendizaje

- La **tasa de aprendizaje**(Learning Rate) puede entenderse como el tamaño del paso en la optimización para minimizar la función de pérdida de la red, es decir, es un parámetro que determina cuánto influye un paso de actualización en el valor actual de los pesos, (Jordan, 2018).
- Por lo que este parámetro suele disminuir gradualmente durante las épocas de la fase de entrenamiento. Si actualizamos todos los pesos usando esta misma fórmula, esto equivale a moverse en la dirección de descenso más pronunciado a lo largo de la superficie de error, de ahí el nombre, descenso de gradiente.



Tasa de Aprendizaje (Learning Rate)

Al entrenar redes neuronales profundas, a menudo es útil reducir la tasa de aprendizaje a medida que avanza el entrenamiento y no mantenerlo constante y así evitar la divergencia (punto a partir del cual la pérdida ya no se reduce y en lugar de eso comienza a incrementarse), Jordan (2018).

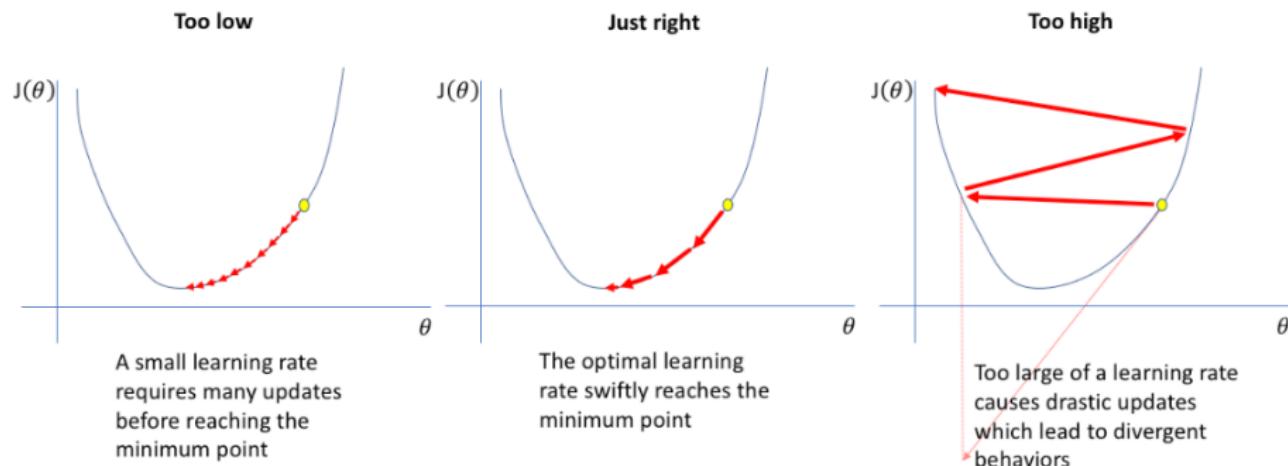


Figura: Establecimiento de la Tasa de Aprendizaje
(Jordan, 2018)



Tasa de Aprendizaje (Learning Rate)

- Reduciendo lentamente la tasa de aprendizaje a lo largo del tiempo ayuda a acelerar el aprendizaje. Esto es lo que se conoce como tasa de decaimiento de aprendizaje (**learning rate decay**).



Tasa de Aprendizaje (Learning Rate)

- Reduciendo lentamente la tasa de aprendizaje a lo largo del tiempo ayuda a acelerar el aprendizaje. Esto es lo que se conoce como tasa de decaimiento de aprendizaje (**learning rate decay**).
- Sin embargo, el optimizador Gradient Descent (en cualquiera de sus tipos) con una tasa de decaimiento de aprendizaje, no es usado a menudo para entrenar redes neuronales profundas ya que la misma tasa de aprendizaje se aplica a todas las actualizaciones de los parámetros.



Tasa de Aprendizaje (Learning Rate)

- Reduciendo lentamente la tasa de aprendizaje a lo largo del tiempo ayuda a acelerar el aprendizaje. Esto es lo que se conoce como tasa de decaimiento de aprendizaje (**learning rate decay**).
- Sin embargo, el optimizador Gradient Descent (en cualquiera de sus tipos) con una tasa de decaimiento de aprendizaje, no es usado a menudo para entrenar redes neuronales profundas ya que la misma tasa de aprendizaje se aplica a todas las actualizaciones de los parámetros.
- Opta en usar optimizadores más avanzados que tienen una tasa de convergencia más rápida, adaptable a diversas situaciones y son conocidos como optimizadores de descenso de gradiente de segundo orden; tales como optimizador Adagrad, Adadelta, RMSprop, Adam, entre otros.



Optimizador ADAM

- Es un algoritmo de optimización de tasa de aprendizaje adaptativo que ha sido diseñado específicamente para el entrenamiento de redes neuronales profundas; el nombre Adam deriva de Estimación del Momento Adaptativo (Kingma and Ba, 2015).



Optimizador ADAM

- Es un algoritmo de optimización de tasa de aprendizaje adaptativo que ha sido diseñado específicamente para el entrenamiento de redes neuronales profundas; el nombre Adam deriva de Estimación del Momento Adaptativo (Kingma and Ba, 2015).
- Permite que la tasa de aprendizaje se adapte según los parámetros y realiza actualizaciones más grandes para parámetros infrecuentes y actualizaciones más pequeñas para los frecuentes.



Optimizador ADAM

- Es un algoritmo de optimización de tasa de aprendizaje adaptativo que ha sido diseñado específicamente para el entrenamiento de redes neuronales profundas; el nombre Adam deriva de Estimación del Momento Adaptativo (Kingma and Ba, 2015).
- Permite que la tasa de aprendizaje se adapte según los parámetros y realiza actualizaciones más grandes para parámetros infrecuentes y actualizaciones más pequeñas para los frecuentes.
- Otra ventaja es que básicamente simplifica la necesidad de ajustar la tasa de aprendizaje debido a que cada parámetro tiene su propia tasa de aprendizaje.



Validación Cruzada

- Para el entrenamiento, el conjunto de datos se divide en un subconjunto de entrenamiento y otro para validación.



Validación Cruzada

- Para el entrenamiento, el conjunto de datos se divide en un subconjunto de entrenamiento y otro para validación.
- El modelo es entrenado sobre el conjunto de entrenamiento completo y la capacidad de generalización es medida en el conjunto de validación.



Validación Cruzada

- Para el entrenamiento, el conjunto de datos se divide en un subconjunto de entrenamiento y otro para validación.
- El modelo es entrenado sobre el conjunto de entrenamiento completo y la capacidad de generalización es medida en el conjunto de validación.
- La validación cruzada es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y validación, (Moore, 2001).



Validación Cruzada

- Para el entrenamiento, el conjunto de datos se divide en un subconjunto de entrenamiento y otro para validación.
- El modelo es entrenado sobre el conjunto de entrenamiento completo y la capacidad de generalización es medida en el conjunto de validación.
- La validación cruzada es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y validación, (Moore, 2001).
- El resultado de la validación cruzada representa la pérdida de la entropía del conjunto de datos. La entropía se suele utilizar en la teoría de la información para medir la pureza o impureza de un conjunto determinado.

$$H(p, q) = - \sum_{\forall x} p(x) \log(q(x))$$

Fórmula de entropía cruzada con dos distribuciones sobre la variable discreta x , donde $q(x)$ es la estimación para la clasificación verdadera $p(x)$



Validación Cruzada

- Para una red neuronal, normalmente verá la ecuación escrita en una forma donde y es el vector de datos reales y la variable x (o algún otro valor tomado directamente de la última salida de la capa) es la estimación, se vería así para un solo ejemplo:

$$L = -\mathbf{y} \cdot \log(\mathbf{x})$$



Validación Cruzada

- Para una red neuronal, normalmente verá la ecuación escrita en una forma donde y es el vector de datos reales y la variable x (o algún otro valor tomado directamente de la última salida de la capa) es la estimación, se vería así para un solo ejemplo:

$$L = -\mathbf{y} \cdot \log(\mathbf{x})$$

- A menudo esta ecuación es promediada sobre todos los ejemplos como una función de costo.

$$J = -\frac{1}{N} \left(\sum_{i=1}^N \mathbf{y}_i \cdot \log(\mathbf{x}_i) \right)$$

Función de costo basada en pérdida de clasificación de múltiples clases para un conjunto de datos de tamaño N



Método de Regularización L2

- La regularización es una técnica que realiza ligeras modificaciones en el algoritmo de aprendizaje, de manera que el modelo se generaliza mejor.



Método de Regularización L2

- La regularización es una técnica que realiza ligeras modificaciones en el algoritmo de aprendizaje, de manera que el modelo se generaliza mejor.
- El método de Regularización L2 o también conocido como Regresión Lasso, trata de agregar un término adicional(λ) a la función de coste, la cual penaliza parametrizaciones con pesos más elevados, es decir, reduce el coeficiente(peso) de las funciones menos importantes a cero eliminando así algunas características, (Romero, 2015b).



Método de Regularización L2

- La regularización es una técnica que realiza ligeras modificaciones en el algoritmo de aprendizaje, de manera que el modelo se generaliza mejor.
- El método de Regularización L2 o también conocido como Regresión Lasso, trata de agregar un término adicional(lambda) a la función de coste, la cual penaliza parametrizaciones con pesos más elevados, es decir, reduce el coeficiente(peso) de las funciones menos importantes a cero eliminando así algunas características, (Romero, 2015b).
- Si $L(\theta, D)$ es la función de coste, θ es el conjunto de parámetros libres y D es un ejemplo de entrenamiento, entonces a la función de pérdida regularizada será:

$$E(\theta, D) = L(\theta, D) + \lambda R(\theta)$$



Método de Regularización L2

- La regularización es una técnica que realiza ligeras modificaciones en el algoritmo de aprendizaje, de manera que el modelo se generaliza mejor.
- El método de Regularización L2 o también conocido como Regresión Lasso, trata de agregar un término adicional(λ) a la función de coste, la cual penaliza parametrizaciones con pesos más elevados, es decir, reduce el coeficiente(peso) de las funciones menos importantes a cero eliminando así algunas características, (Romero, 2015b).
- Si $L(\theta, D)$ es la función de coste, θ es el conjunto de parámetros libres y D es un ejemplo de entrenamiento, entonces a la función de pérdida regularizada será:

$$E(\theta, D) = L(\theta, D) + \lambda R(\theta)$$

- En esta investigación se usó una constante **$\lambda = 0.0001$** . La regularización L2 incluye los pesos de las capas totalmente conectadas y convolucionales, pero no incluye a los sesgos (biases).

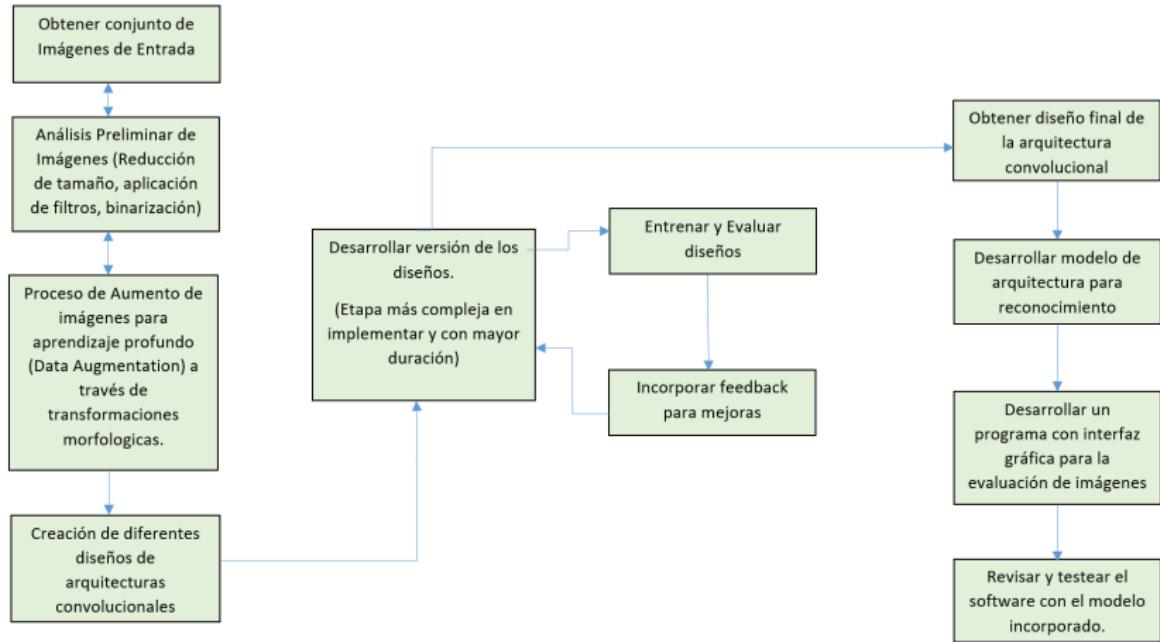


Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

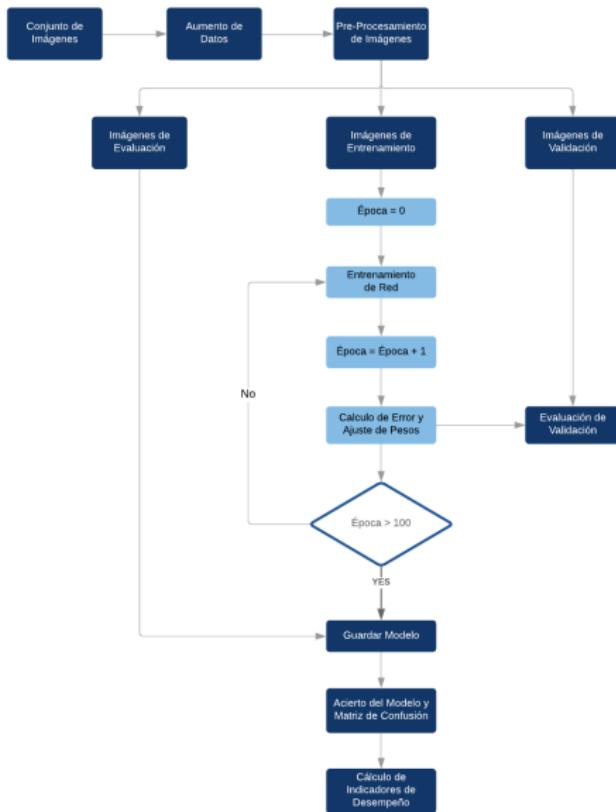
- 1 Introducción
- 2 Motivación
- 3 Formulación del problema e Hipótesis
- 4 Importancia de la investigación
- 5 Contribución de la investigación
- 6 Marco teórico
 - Aprendizaje Profundo
 - Red Convolutacional
 - Arquitectura del Modelo
 - Componentes del Modelo
 - Entrenamiento y Validación
- 7 Propuesta o tema central de la tesis
- 8 Resultados de la tesis
- 9 Consideraciones finales
- 10 Referencias bibliograficas



Diseño del Ciclo de Vida del Desarrollo del Modelo

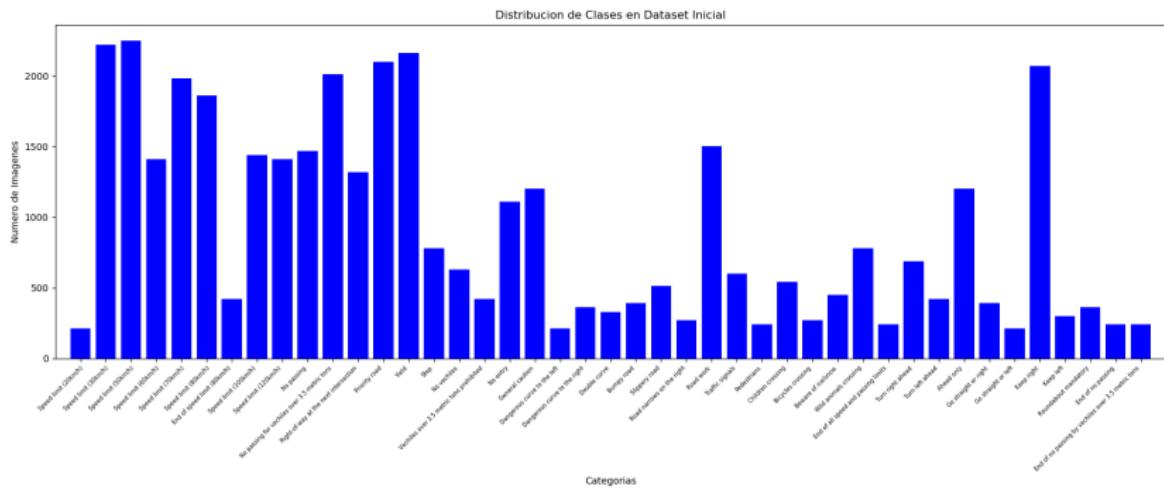


Proceso de Desarrollo del Modelo



Datos para el entrenamiento

• Señales de Tránsito de Alemania

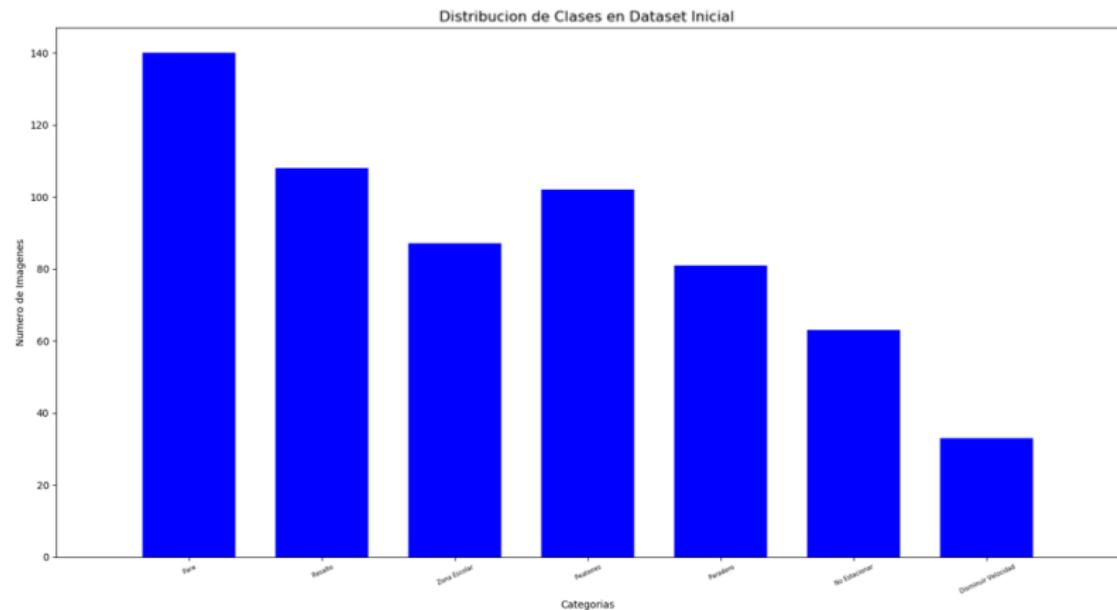


Distribución de ejemplos por señal para el entrenamiento (Total 39209) Fuente: Elaboración propia



Datos para el entrenamiento

- Señales de Tránsito de Perú

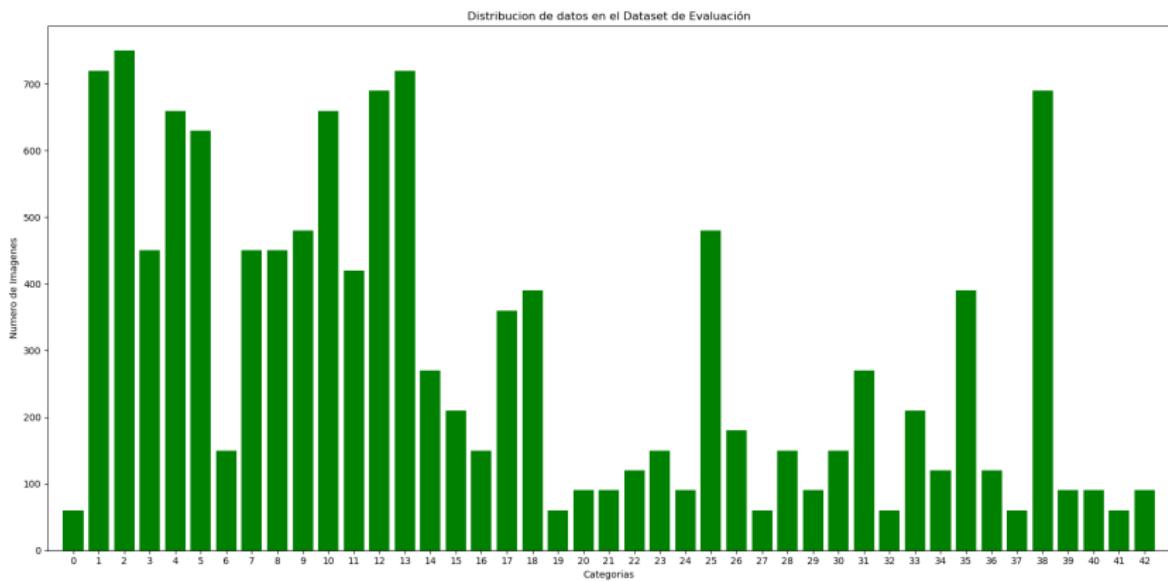


Distribución de ejemplos por señal para el entrenamiento(Total 614) Fuente: Elaboración propia



Datos para la Evaluación

- Señales de Tránsito de Alemania (12630 imágenes)

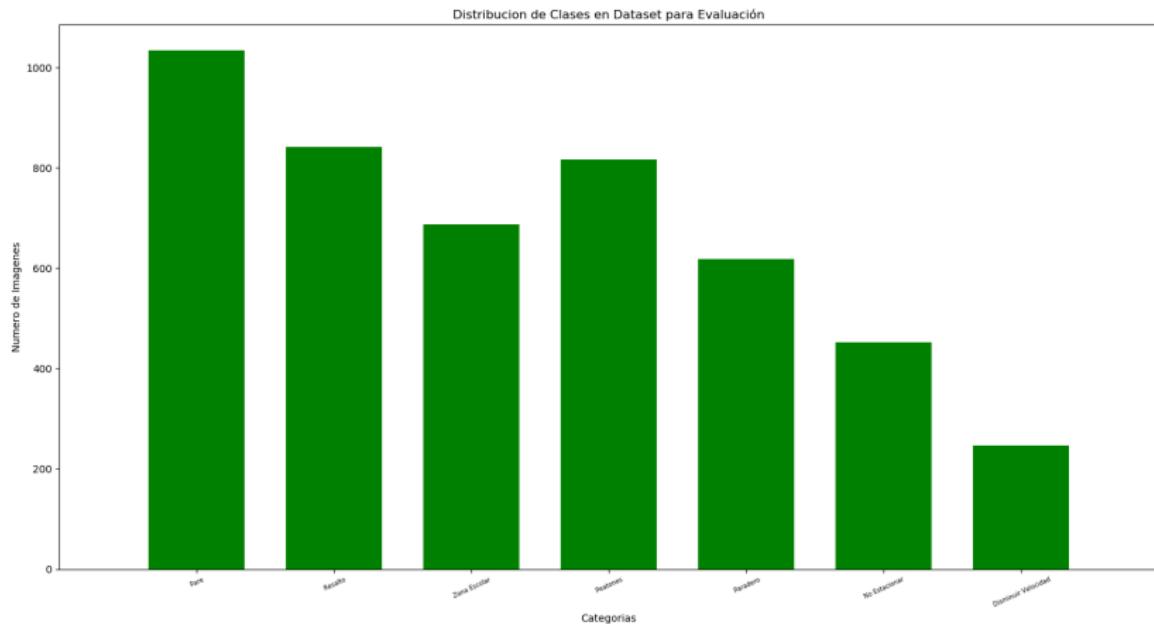


Distribución de ejemplos por señal para la evaluación - Alemania Fuente: Elaboración propia



Datos para la Evaluación

- Señales de Tránsito de Perú (4698 imágenes)



Distribución de ejemplos por señal para la evaluación - Perú Fuente: Elaboración propia



Proceso de Aumento de Datos(Data Augmentation)

① Flipping

Flipping horizontal:



Fuente: Elaboración propia

Flipping Vertical:

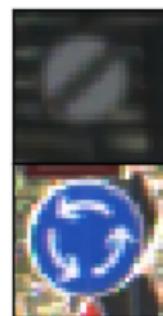
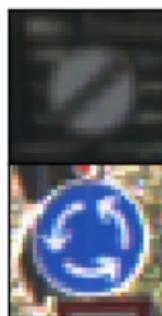


Fuente: Elaboración propia



Proceso de Aumento de Datos(Data Augmentation)

Flipping Horizontal y Vertical:



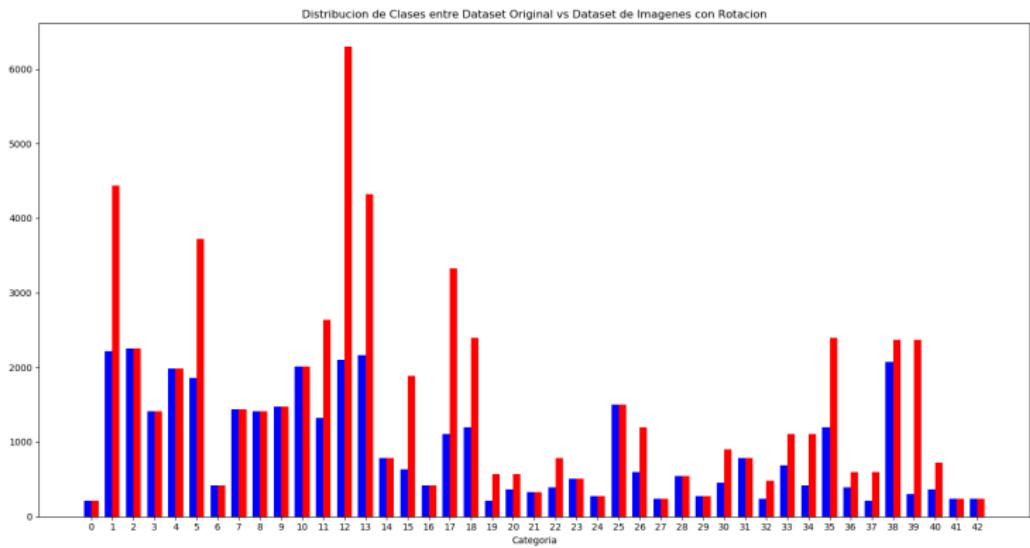
Proceso de Aumento de Datos(Data Augmentation)

Incluso, hay signos que luego de voltearse, deben clasificarse como un signo de alguna otra clase. Esto sigue siendo útil, ya que podemos utilizar los datos de estas clases para ampliar sus contrapartes.



Proceso de Aumento de Datos(Data Augmentation)

Finalmente obtenemos una nueva distribución de datos luego de haber aplicado flipping a ciertas imágenes. Esta distribución consta de 63538 imágenes.



Distribución de categoría, luego de aplicar el Flip(en sus distintos tipos)- Señales de Alemania

Fuente: Elaboración propia



Proceso de Aumento de Datos(Data Augmentation)

② Projection(Proyección)



Ejemplo de cinco proyecciones por cada imagen - Dataset Perú

Fuente: Elaboración propia



Proceso de Aumento de Datos(Data Augmentation)

③ Rotation(Rotación)



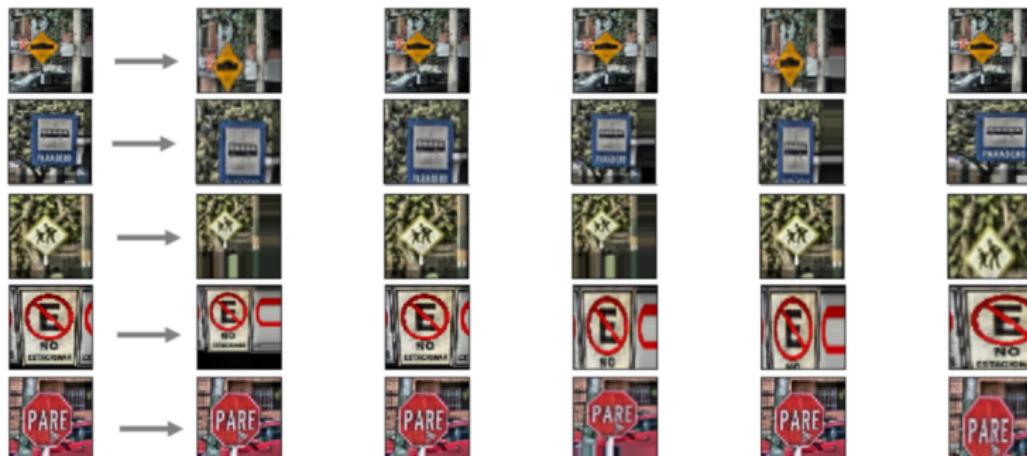
Ejemplo de cinco rotaciones por cada imagen - Dataset Perú

Fuente: Elaboración propia



Proceso de Aumento de Datos(Data Augmentation)

4 Zoom



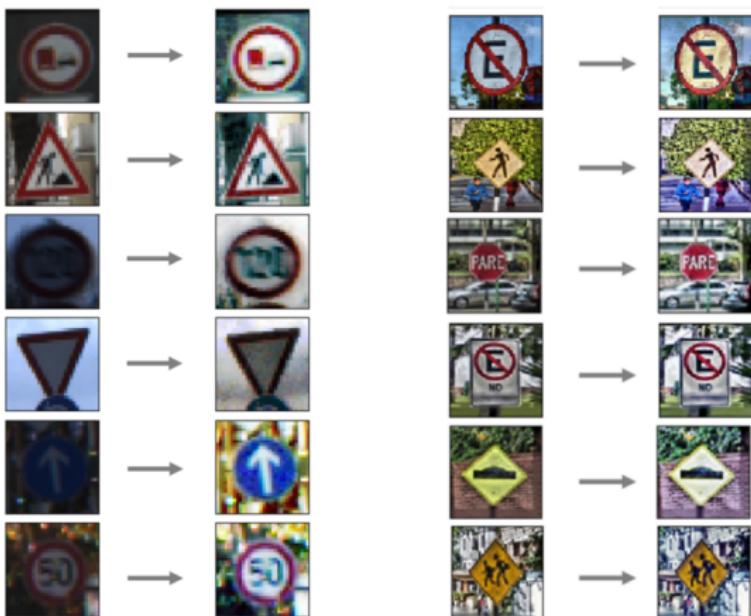
Ejemplo de cinco aplicaciones de zoom(in/out) por cada imagen - Dataset Perú

Fuente: Elaboración propia



Proceso de Aumento de Datos(Data Augmentation)

5 Equalizacion del histograma



Distribución eficaz de los valores de intensidad más frecuentes

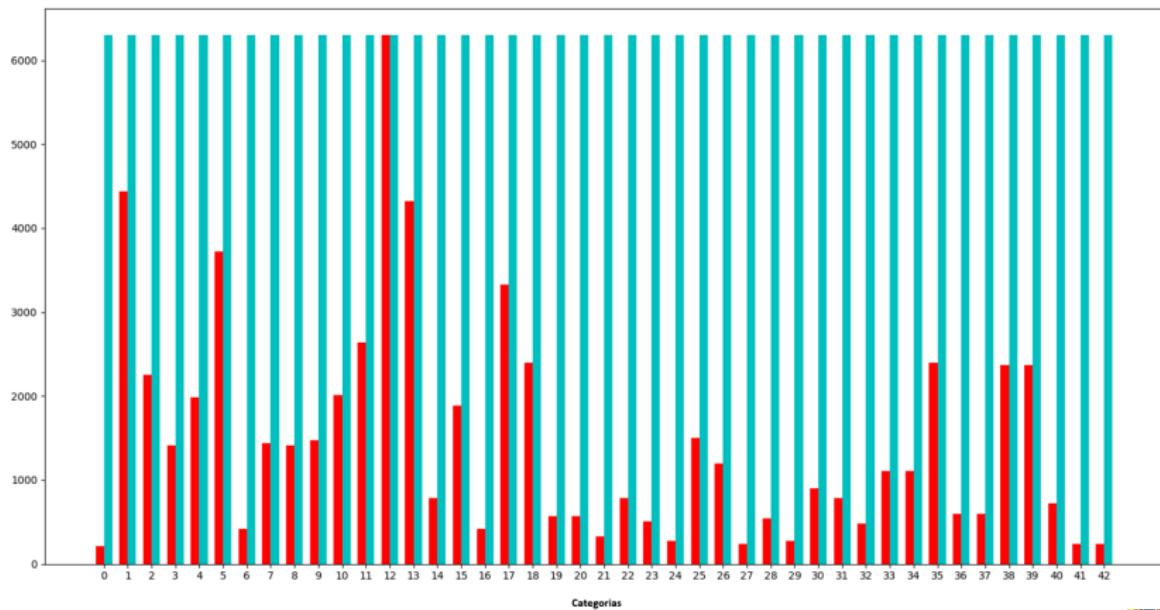
Fuente: Elaboración propia



Dataset final

- Señales de Tránsito de Alemania (270900 imágenes)

Distribución de Clases Imágenes Rotadas vs Imágenes Aumentadas manteniendo un Balance

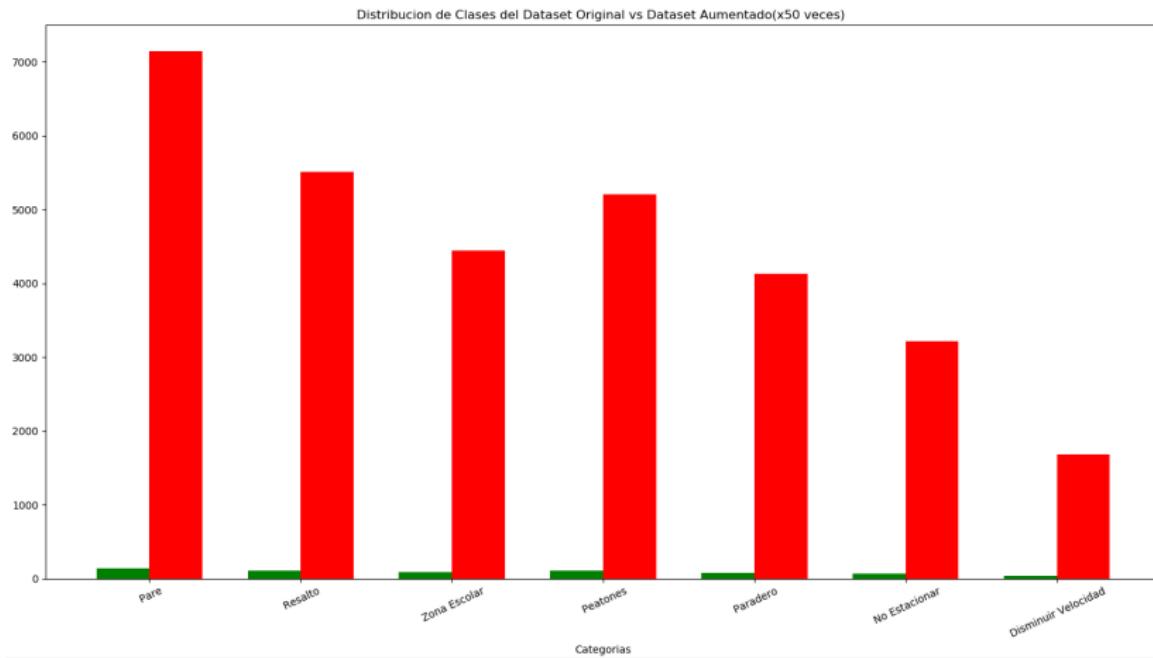


Fuente: Elaboración propia



Dataset final

- Señales de Tránsito de Perú (31314 imágenes)



Fuente: Elaboración propia



Dataset final

Tabla: Distribución Entrenamiento y Validación Dataset Balanceado - Señales de Tránsito de Alemania

CONJUNTO DE DATOS	CANTIDAD IMÁGENES
Entrenamiento	203175
Validación	67725
Evaluación	12630

Fuente: Elaboración propia.

Tabla: Distribución Entrenamiento y Validación Dataset no Balanceado - Señales de Tránsito de Perú

CONJUNTO DE DATOS	CANTIDAD IMÁGENES
Entrenamiento	23485
Validación	3131
Evaluación	4698

Fuente: Elaboración propia.



Pre-procesamiento de Imágenes(Normalization)

- La técnica utilizada CLAHE es un algoritmo para la mejora del contraste local, que utiliza histogramas calculados sobre diferentes regiones en una imagen.



Pre-procesamiento de Imágenes(Normalization)

- La técnica utilizada CLAHE es un algoritmo para la mejora del contraste local, que utiliza histogramas calculados sobre diferentes regiones en una imagen.
- Difiere de la ecualización de histograma ordinaria en el sentido de que el método adaptativo computa varios histogramas, cada uno correspondiente a una sección distinta de la imagen, y los utiliza para redistribuir los valores de luminosidad de la imagen.



Pre-procesamiento de Imágenes(Normalization)

- La técnica utilizada CLAHE es un algoritmo para la mejora del contraste local, que utiliza histogramas calculados sobre diferentes regiones en una imagen.
- Difiere de la ecualización de histograma ordinaria en el sentido de que el método adaptativo computa varios histogramas, cada uno correspondiente a una sección distinta de la imagen, y los utiliza para redistribuir los valores de luminosidad de la imagen.
- Mejora el nivel de visibilidad de una imagen o video con niebla ya que permite mejorar los detalles locales incluso en regiones que son más oscuras o más claras que la mayoría de regiones de la imagen. (Yadav et al., 2014)



Pre-procesamiento de Imágenes(Normalization)

- Pierre Sermanet y Yann LeCun mencionaron en su artículo (LeCun et al., 1998), que el uso de canales de color no pareció mejorar mucho las cosas. Además, debido a diversas condiciones o problemas de iluminación, no es adecuado procesar directamente las imágenes que se capturan a través de la cámara o sensores de imágenes, es por ello que en esta investigación **se usará un solo canal** en el modelo, es decir las imágenes estarán en escala de grises en lugar de tener 3 canales de colores.



Arquitectura del Modelo

- Los hiperparámetros engloban funciones, variables y constantes utilizadas durante la construcción de las diferentes arquitecturas; estas varían, sin embargo siguiendo conceptos teóricos, antecedentes (investigaciones previas) y sobretodo después de algunas pruebas realizadas, los siguientes hiperparámetros fueron seleccionados de manera específica:

Tabla: Hiperparámetros del Modelo

HIPERPARÁMETROS	TIPO	HIPERPARÁMETROS	TIPO
Tasa de Aprendizaje - Alemania	0.0005 (5e-4)	Tasa de Aprendizaje - Perú	0.0001 (1e-4)
Alg. de Optimización	Optimizador Adam	Método de Validación	Entropía Cruzada
Fun. Activ. Capas Convolucionales	RELU y DropOut	Fun. Activ. Capas Totalmente Conectadas	Func. Softmax
Método de Regularización	$L2\ Lasso(\lambda = 0.0001)$	Épocas	100

Fuente: Elaboración propia.



Modelos Propuestos

1 MODELO A

Alemania

Tabla: Diseño A (19'057952 neuronas) - Dataset de Imágenes de Alemania

Capa	Entrada	Tipo	Número de (kernels/filtros)	Padding	Salida	Func.Esc.Múltiple
1	1 de 32 x 32	Conv(DropOut : 0.8)	32 de 3 x 3 neuronas	Activo	32 de 32 x 32	–
	32 de 32 x 32	Max Pool	32 de 2 x 2	Inactivo	32 de 16 x 16	(MaxPool = 2) 32 de 8x8
2	32 de 16 x 16	Conv(DropOut : 0.7)	64 de 5 x 5 neuronas	Activo	64 de 16 x 16	–
	64 de 16 x 16	Max Pool	64 de 2 x 2	Inactivo	64 de 8 x 8	(MaxPool = 1) 64 de 8x8
3	6144 neuronas	F.C.(DropOut : 0.5)	3072 neuronas	–	43	–

Fuent0e: Elaboración propia.



Modelos Propuestos

1 MODELO A

Perú

Tabla: Diseño A (66'428192 neuronas) - Dataset de Imágenes de Perú

Capa	Entrada	Tipo	Número de (kernels/filtros)	Padding	Salida	Func.Esc.Múltiple
1	1 de 60 x 60	Conv(DropOut : 0.8)	32 de 3 x 3 neuronas	Activo	32 de 60 x 60	–
	32 de 60 x 60	Max Pool	32 de 2 x 2	Inactivo	32 de 30 x 30	(MaxPool = 2) 32 de 15x15
2	32 de 30 x 30	Conv(DropOut : 0.7)	64 de 5 x 5 neuronas	Activo	64 de 30 x 30	–
	64 de 30 x 30	Max Pool	64 de 2 x 2	Inactivo	64 de 15 x 15	(MaxPool = 1) 64 de 15x15
3	21600 neuronas	F.C.(DropOut : 0.5)	7200 neuronas	–	7	–

Fuente: Elaboración propia.



Modelos Propuestos

② MODELO B

Alemania

Tabla: Diseño B (3'970336 neuronas) - Dataset de Imágenes de Alemania

Capa	Entrada	Tipo	Número de (kernels/filtros)	Padding	Salida	Func.Esc.Múltiple
1	1 de 32 x 32	Conv(DropOut : 0.8)	32 de 3 x 3 neuronas	Activo	32 de 32 x 32	–
	32 de 32 x 32	Max Pool	32 de 2 x 2	Inactivo	32 de 16 x 16	(MaxPool = 4) 32 de 4x4
2	32 de 16 x 16	Conv(DropOut : 0.7)	64 de 5 x 5 neuronas	Activo	64 de 16 x 16	–
	64 de 16 x 16	Max Pool	64 de 2 x 2	Inactivo	64 de 8 x 8	(MaxPool = 2) 64 de 4x4
3	64 de 8 x 8	Conv(DropOut : 0.6)	128 de 5 x 5 neuronas	Activo	128 de 8 x 8	–
	128 de 8 x 8	Max Pool	128 de 2 x 2	Inactivo	128 de 4 x 4	(MaxPool = 1) 128 de 4x4
4	3584 neuronas	F.C.(DropOut : 0.5)	1024 neuronas	–	43	–

Fuente: Elaboración propia.



Modelos Propuestos

② MODELO B

Perú

Tabla: Diseño B (29'910388 neuronas) - Dataset de Imágenes de Perú

Capa	Entrada	Tipo	Número de (kernels/filtros)	Padding	Salida	Func.Esc.Múltiple
1	1 de 60 x 60	Conv(DropOut : 0.8)	32 de 3 x 3 neuronas	Activo	32 de 60 x 60	–
	32 de 60 x 60	Max Pool	32 de 2 x 2	Inactivo	32 de 30 x 30	(MaxPool = 4) 32 de 7x7
2	32 de 30 x 30	Conv(DropOut : 0.7)	64 de 5 x 5 neuronas	Activo	64 de 30 x 30	–
	64 de 30 x 30	Max Pool	64 de 2 x 2	Inactivo	64 de 15 x 15	(MaxPool = 2) 64 de 7x7
3	64 de 15 x 15	Conv(DropOut : 0.6)	128 de 5 x 5 neuronas	Activo	128 de 15 x 15	–
	128 de 15 x 15	Max Pool	128 de 2 x 2	Inactivo	128 de 7 x 7	(MaxPool = 1) 128 de 7x7
4	10976 neuronas	F.C.(DropOut : 0.5)	2700 neuronas	–	7	–

Fuente: Elaboración propia.



Modelos Propuestos

8 MODELO C

Alemania

Tabla: Diseño C (3'937568 neuronas) - Dataset de Imágenes de Alemania

Capa	Entrada	Tipo	Número de (kernels/filtros)	Padding	Salida	Func.Esc.Múltiple
1	1 de 32 x 32	Conv(DropOut : 0.8)	32 de 3 x 3 neuronas	Activo	32 de 32 x 32	–
	32 de 32 x 32	Max Pool	32 de 2 x 2	Inactivo	32 de 16 x 16	(MaxPool = 4) 32 de 4x4
2	32 de 16 x 16	Conv(DropOut : 0.7)	64 de 3 x 3 neuronas	Activo	64 de 16 x 16	–
	64 de 16 x 16	Max Pool	64 de 2 x 2	Inactivo	64 de 8 x 8	(MaxPool = 2) 64 de 4x4
3	64 de 8 x 8	Conv(DropOut : 0.6)	128 de 5 x 5 neuronas	Activo	128 de 8 x 8	–
	128 de 8 x 8	Max Pool	128 de 2 x 2	Inactivo	128 de 4 x 4	(MaxPool = 1) 128 de 4x4
4	3584 neuronas	F.C.(DropOut : 0.5)	1024 neuronas	–	43	–

Fuente: Elaboración propia.



Modelos Propuestos

8 MODELO C

Perú

Tabla: Diseño C (29'877620 neuronas) - Dataset de Imágenes de Perú

Capa	Entrada	Tipo	Número de (kernels/filtros)	Padding	Salida	Func.Esc.Múltiple
1	1 de 60 x 60	Conv(DropOut : 0.8)	32 de 3 x 3 neuronas	Activo	32 de 60 x 60	–
	32 de 60 x 60	Max Pool	32 de 2 x 2	Inactivo	32 de 30 x 30	(MaxPool = 4) 32 de 7x7
2	32 de 30 x 30	Conv(DropOut : 0.7)	64 de 3 x 3 neuronas	Activo	64 de 30 x 30	–
	64 de 30 x 30	Max Pool	64 de 2 x 2	Inactivo	64 de 15 x 15	(MaxPool = 2) 64 de 7x7
3	64 de 15 x 15	Conv(DropOut : 0.6)	128 de 5 x 5 neuronas	Activo	128 de 15 x 15	–
	128 de 15 x 15	Max Pool	128 de 2 x 2	Inactivo	128 de 7 x 7	(MaxPool = 1) 128 de 7x7
4	10976 neuronas	F.C.(DropOut : 0.5)	2700 neuronas	–	7	–

Fuente: Elaboración propia.



Modelos Propuestos

4 MODELO D

Alemania

Tabla: Diseño D (4'166944 neuronas) - Dataset de Imágenes de Alemania

Capa	Entrada	Tipo	Número de (kernels/filtros)	Padding	Salida	Func.Esc.Múltiple
1	1 de 32 x 32	Conv(DropOut : 0.8)	32 de 3 x 3 neuronas	Activo	32 de 32 x 32	–
	32 de 32 x 32	Max Pool	32 de 2 x 2	Inactivo	32 de 16 x 16	(MaxPool = 4) 32 de 4x4
2	32 de 16 x 16	Conv(DropOut : 0.7)	64 de 5 x 5 neuronas	Activo	64 de 16 x 16	–
	64 de 16 x 16	Max Pool	64 de 2 x 2	Inactivo	64 de 8 x 8	(MaxPool = 2) 64 de 4x4
3	64 de 8 x 8	Conv(DropOut : 0.6)	128 de 7 x 7 neuronas	Activo	128 de 8 x 8	–
	128 de 8 x 8	Max Pool	128 de 2 x 2	Inactivo	128 de 4 x 4	(MaxPool = 1) 128 de 4x4
4	3584 neuronas	F.C.(DropOut : 0.5)	1024 neuronas	–	43	–

Fuente: Elaboración propia.



Modelos Propuestos

4 MODELO D

Perú

Tabla: Diseño D (30'106996 neuronas) - Dataset de Imágenes de Perú

Capa	Entrada	Tipo	Número de (kernels/filtros)	Padding	Salida	Func.Esc.Múltiple
1	1 de 60 x 60	Conv(DropOut : 0.8)	32 de 3 x 3 neuronas	Activo	32 de 60 x 60	–
	32 de 60 x 60	Max Pool	32 de 2 x 2	Inactivo	32 de 30 x 30	(MaxPool = 4) 32 de 7x7
2	32 de 30 x 30	Conv(DropOut : 0.7)	64 de 5 x 5 neuronas	Activo	64 de 30 x 30	–
	64 de 30 x 30	Max Pool	64 de 2 x 2	Inactivo	64 de 15 x 15	(MaxPool = 2) 64 de 7x7
3	64 de 15 x 15	Conv(DropOut : 0.6)	128 de 7 x 7 neuronas	Activo	128 de 15 x 15	–
	128 de 15 x 15	Max Pool	128 de 2 x 2	Inactivo	128 de 7 x 7	(MaxPool = 1) 128 de 7x7
4	10976 neuronas	F.C.(DropOut : 0.5)	2700 neuronas	–	7	–

Fuente: Elaboración propia.



Modelos Propuestos

5 MODELO E

Alemania

Tabla: Diseño E (2'077706 neuronas) - Dataset de Imágenes de Alemania

Capa	Entrada	Tipo	Número de (kernels/filtros)	Padding	Salida	Func.Esc.Múltiple
1	1 de 32 x 32	Conv(DropOut : 0.8)	32 de 3 x 3 neuronas	Activo	32 de 32 x 32	–
	32 de 32 x 32	Max Pool	32 de 2 x 2	Inactivo	32 de 16 x 16	(MaxPool = 8) 32 de 2x2
2	32 de 16 x 16	Conv(DropOut : 0.7)	64 de 5 x 5 neuronas	Activo	64 de 16 x 16	–
	64 de 16 x 16	Max Pool	64 de 2 x 2	Inactivo	64 de 8 x 8	(MaxPool = 4) 64 de 2x2
3	64 de 8 x 8	Conv(DropOut : 0.6)	128 de 5 x 5 neuronas	Activo	128 de 8 x 8	–
	128 de 8 x 8	Max Pool	128 de 2 x 2	Inactivo	128 de 4 x 4	(MaxPool = 2) 128 de 2x2
4	128 de 4 x 4	Conv(DropOut : 0.6)	128 de 7 x 7 neuronas	Activo	128 de 4 x 4	–
	128 de 4 x 4	Max Pool	128 de 2 x 2	Inactivo	128 de 2 x 2	(MaxPool = 1) 128 de 2x2
5	1408 neuronas	F.C.(DropOut : 0.5)	702 neuronas	–	43	–

Fuente: Elaboración propia.



Modelos Propuestos

5 MODELO E

Perú

Tabla: Diseño E (6'088304 neuronas) - Dataset de Imágenes de Perú

Capa	Entrada	Tipo	Número de (kernels/filtros)	Padding	Salida	Func.Esc.Múltiple
1	1 de 60 x 60	Conv(DropOut : 0.8)	32 de 3 x 3 neuronas	Activo	32 de 60 x 60	–
	32 de 60 x 60	Max Pool	32 de 2 x 2	Inactivo	32 de 30 x 30	(MaxPool = 8) 32 de 3x3
2	32 de 30 x 30	Conv(DropOut : 0.7)	64 de 5 x 5 neuronas	Activo	64 de 30 x 30	–
	64 de 30 x 30	Max Pool	64 de 2 x 2	Inactivo	64 de 15 x 15	(MaxPool = 4) 64 de 3x3
3	64 de 15 x 15	Conv(DropOut : 0.6)	128 de 5 x 5 neuronas	Activo	128 de 15 x 15	–
	128 de 15 x 15	Max Pool	128 de 2 x 2	Inactivo	128 de 7 x 7	(MaxPool = 2) 128 de 3x3
4	128 de 7 x 7	Conv(DropOut : 0.6)	128 de 7 x 7 neuronas	Activo	128 de 7 x 7	–
	128 de 7 x 7	Max Pool	128 de 2 x 2	Inactivo	128 de 3 x 3	(MaxPool = 1) 128 de 3x3
5	3168 neuronas	F.C.(DropOut : 0.5)	1584 neuronas	–	7	–

Fuente: Elaboración propia.



Resultados del Entrenamiento - Dataset Alemania

Los 5 modelos lograron similares resultados de acierto durante el entrenamiento.

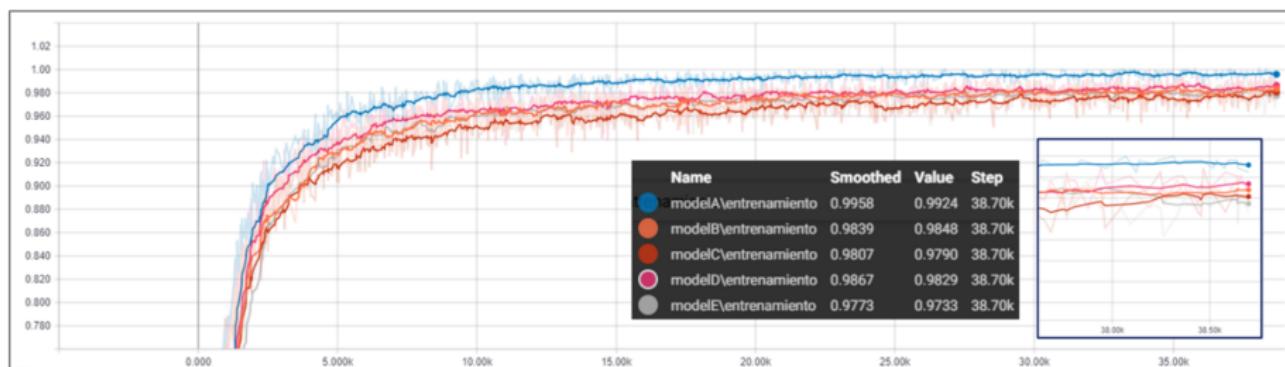


Figura: Análisis del acierto en el Entrenamiento de los modelos - Dataset Señales de Tránsito de Alemania



Resultados del Entrenamiento - Dataset Alemania

Los 5 modelos lograron similares resultados de error durante el entrenamiento.

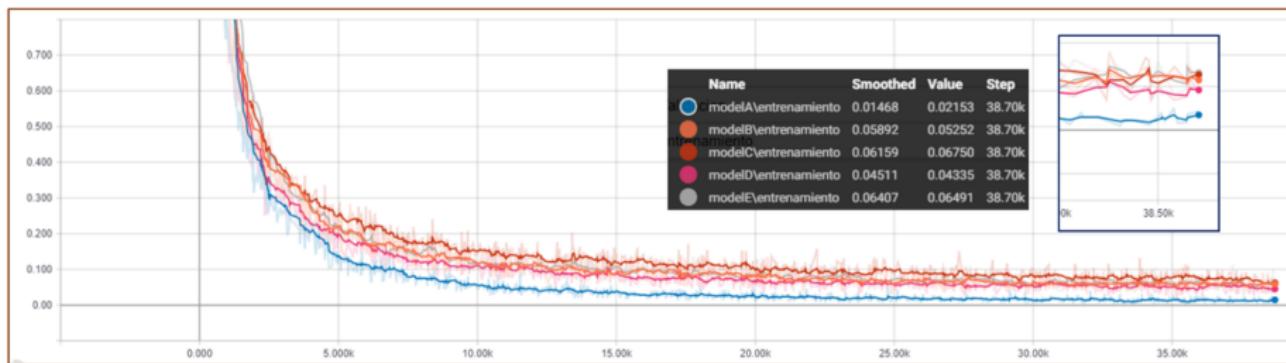


Figura: Análisis del error en el Entrenamiento de los modelos - Dataset Señales de Tránsito de Alemania



Resultados de Validación - Dataset Alemania

El Modelo E obtuvo una de las mejores tasas de Acierto.

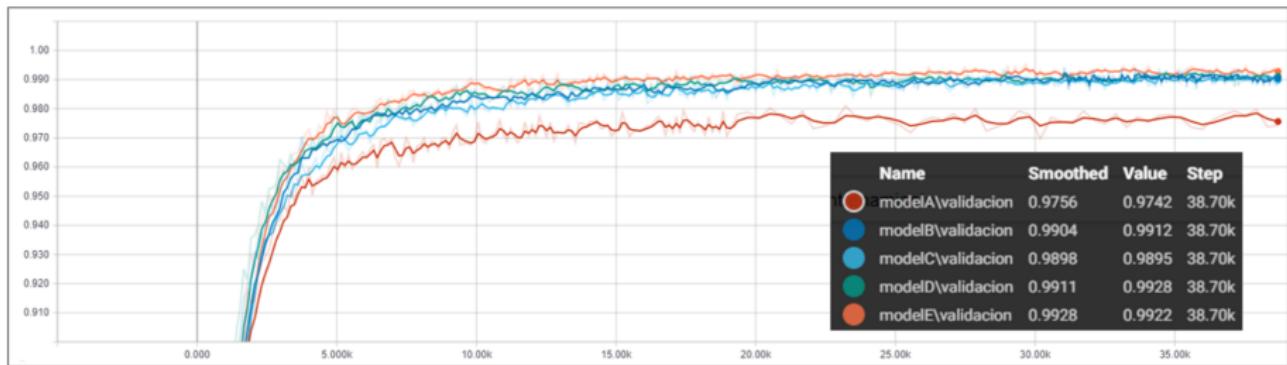


Figura: Análisis del acierto en la Validación de los modelos - Dataset Señales de Tránsito de Alemania



Resultados de Validación - Dataset Alemania

El Modelo E obtuvo una de las mejores tasas de Acierto.

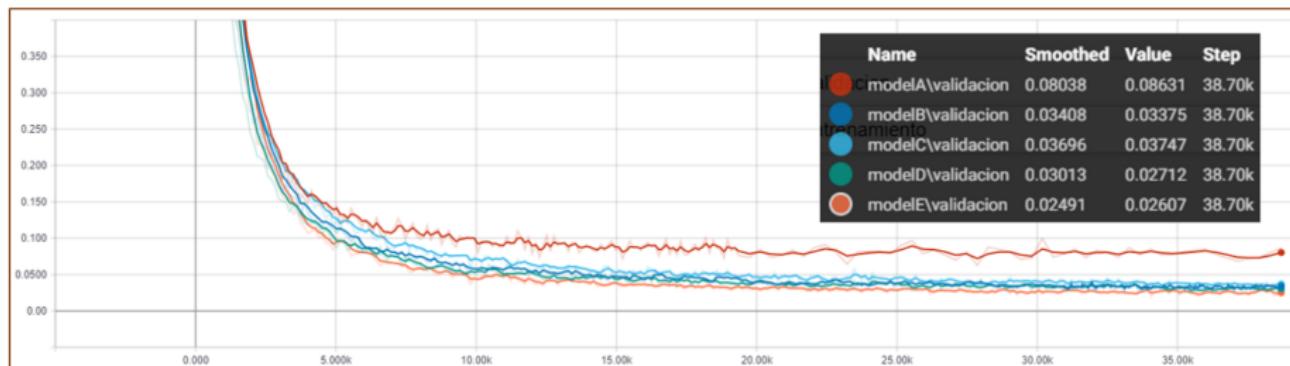


Figura: Análisis del error en la Validación de los modelos - Dataset Señales de Tránsito de Alemania



Resultados del Entrenamiento - Dataset Perú

Los 5 modelos lograron similares resultados de acierto durante el entrenamiento.

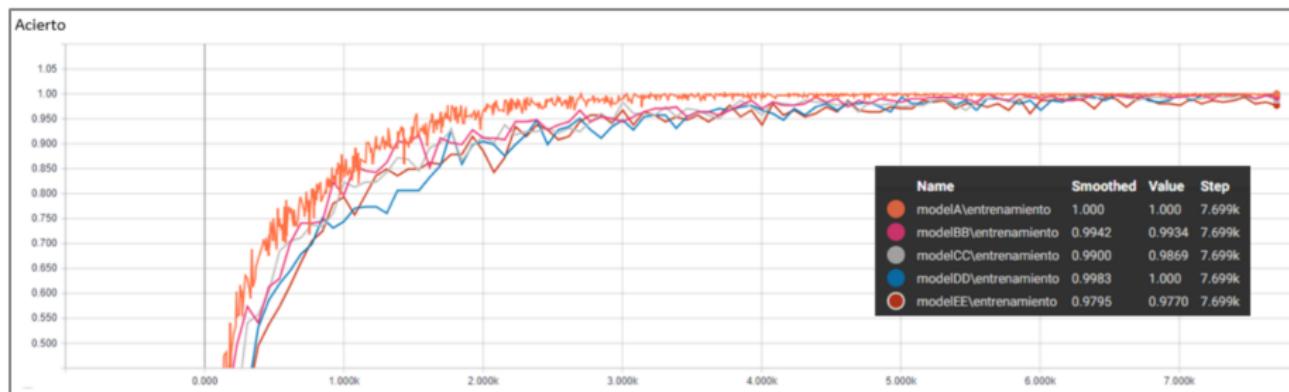


Figura: Análisis del Acierto en el Entrenamiento de los modelos - Dataset Señales de Tránsito de Perú



Resultados del Entrenamiento - Dataset Perú

Los 5 modelos lograron similares resultados de error durante el entrenamiento.

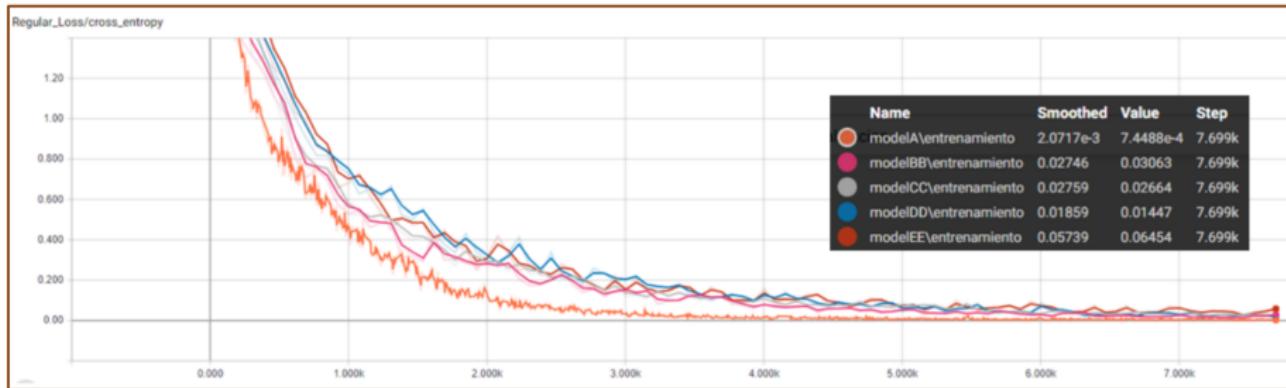


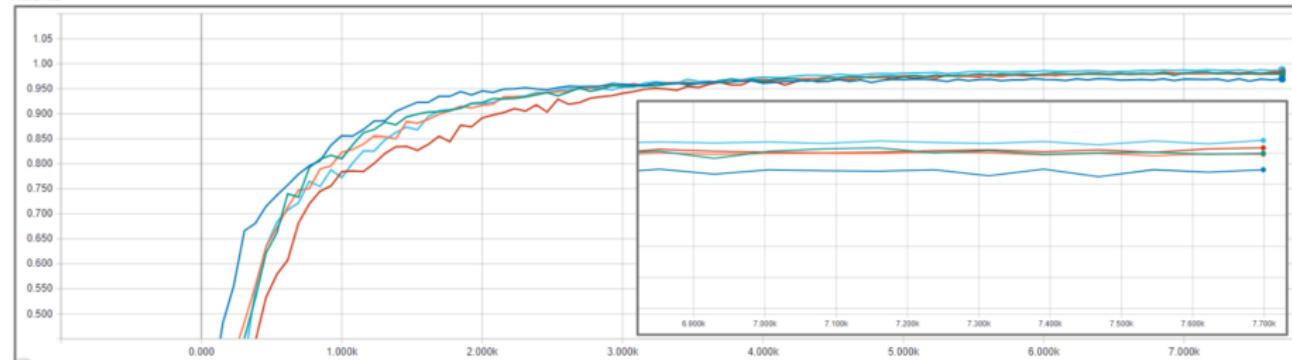
Figura: Análisis del Error en el Entrenamiento de los modelos - Dataset Señales de Tránsito de Perú



Resultados de Validación - Dataset Perú

El Modelo E obtuvo una de las mejores tasas de Acierto.

Acierto



Name	Smoothed	Value	Step
modelA\validacion	0.9693	0.9693	7.699k
modelBB\validacion	0.9799	0.9799	7.699k
modelCC\validacion	0.9792	0.9792	7.699k
modelDD\validacion	0.9834	0.9834	7.699k
modelEE\validacion	0.9882	0.9882	7.699k

Figura: Análisis del acierto en la Validación de los modelos - Dataset Señales de Tránsito de Perú



Resultados de Validación - Dataset Perú

El Modelo E obtuvo una de las mejores tasas de Error.

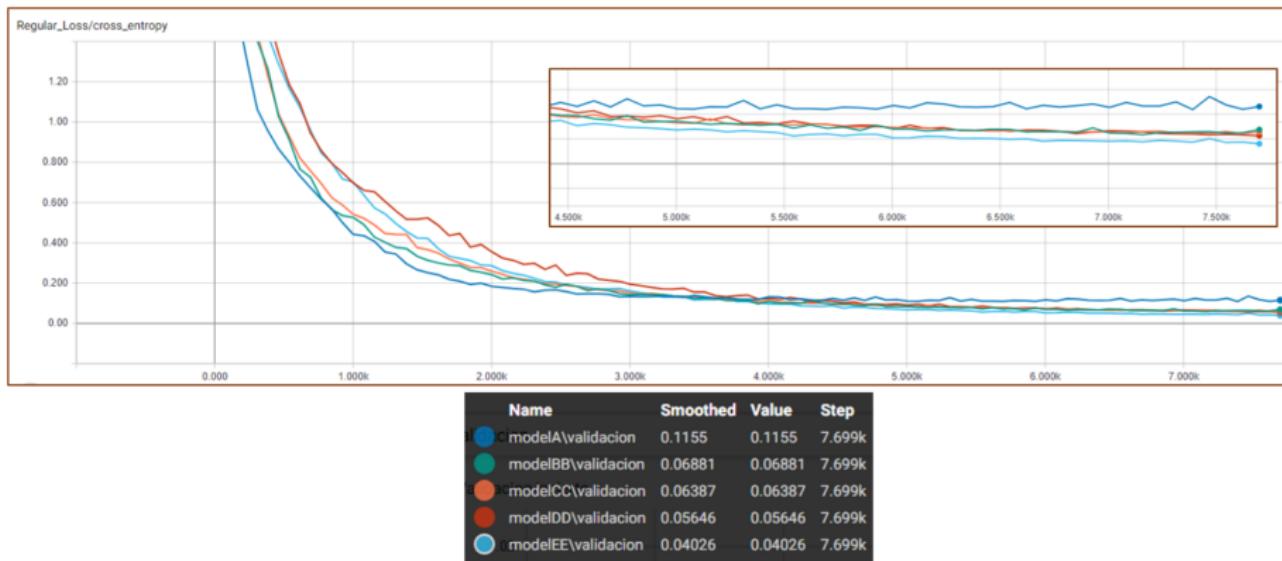


Figura: Análisis del error en la Validación de los modelos - Dataset Señales de Tránsito de Perú



Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

- 1 Introducción
- 2 Motivación
- 3 Formulación del problema e Hipótesis
- 4 Importancia de la investigación
- 5 Contribución de la investigación
- 6 Marco teórico
 - Aprendizaje Profundo
 - Red Convolutacional
 - Arquitectura del Modelo
 - Componentes del Modelo
 - Entrenamiento y Validación
- 7 Propuesta o tema central de la tesis
- 8 **Resultados de la tesis**
- 9 Consideraciones finales
- 10 Referencias bibliograficas



Resultados de la tesis

- Indicadores:

- ▶ Efectividad(Prop. Verdaderos Pos): $PVP = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Negativos}}$
- ▶ Especificidad(Prop. Verdaderos Neg): $PVN = \frac{\text{Verdaderos Negativos}}{\text{Verdaderos Negativos} + \text{Falsos Positivos}}$
- ▶ Valor Predictivo Positivo (Precisión): $PPV = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Positivos}}$
- ▶ Acuracia (Exactitud): $ACC = \frac{\text{Verdaderos Positivos} + \text{Verdaderos Negativos}}{\text{Total de Imagenes}}$



Resultados de la tesis

- Indicadores:

- ▶ Efectividad(Prop. Verdaderos Pos): $PVP = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Negativos}}$
- ▶ Especificidad(Prop. Verdaderos Neg): $PVN = \frac{\text{Verdaderos Negativos}}{\text{Verdaderos Negativos} + \text{Falsos Positivos}}$
- ▶ Valor Predictivo Positivo (Precisión): $PPV = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Positivos}}$
- ▶ Acuracia (Exactitud): $ACC = \frac{\text{Verdaderos Positivos} + \text{Verdaderos Negativos}}{\text{Total de Imágenes}}$

- Donde:

- Verdaderos Positivos: Imágenes correctamente identificadas.
- Falsos Positivos: Imágenes incorrectamente identificadas.
- Verdaderos Negativo: Imágenes correctamente rechazadas.
- Falsos Negativos: Imágenes incorrectamente rechazadas.



Resultados de la tesis

- Utilizando los anteriores indicadores, podemos obtener dos más:
 - ▶ Curvas ROC : Relación entre Efectividad y Especificidad
 - ▶ Curvas PR : Relación entre Precisión y Efectividad(Recall)



Resultados de la tesis

- Utilizando los anteriores indicadores, podemos obtener dos más:
 - ▶ Curvas ROC : Relación entre Efectividad y Especificidad
 - ▶ Curvas PR : Relación entre Precisión y Efectividad(Recall)
- Ambas curvas se utilizan normalmente para estudiar la salida de un clasificador. En las Curvas ROC un aumento en la sensibilidad está acompañado por un decremento en la especificidad. Esto significa que la esquina superior izquierda del gráfico es el punto ideal, debido a que muestran la relación entre las muestras clasificadas adecuadamente (*Proporción de Verdaderos Positivos - PVP*) y las muestras que no pertenecen a la clase pero se clasificaron como si lo fueran (*Proporción de Falsos Positivos - PFP*).



Resultados de la tesis

- De manera similar, las Curvas PR (*Precision -Recall*) representan la relación entre la Efectividad y Precisión del clasificador. El objetivo es tener un modelo que se posicione en la esquina superior derecha, que básicamente consiste en obtener solo los positivos verdaderos sin falsos positivos ni falsos negativos: un clasificador perfecto. En la situación en la que se tiene clases con cantidades no balanceadas, como es el caso del Dataset- Perú, a menudo es más útil utilizar el Área bajo la Curva PR como indicador del clasificador, (Davis and Goadrich, 2006).



Resultados de la tesis

- De manera similar, las Curvas PR (*Precision -Recall*) representan la relación entre la Efectividad y Precisión del clasificador. El objetivo es tener un modelo que se posicione en la esquina superior derecha, que básicamente consiste en obtener solo los positivos verdaderos sin falsos positivos ni falsos negativos: un clasificador perfecto. En la situación en la que se tiene clases con cantidades no balanceadas, como es el caso del Dataset- Perú, a menudo es más útil utilizar el Área bajo la Curva PR como indicador del clasificador, (Davis and Goadrich, 2006).
- El mejor sistema de clasificación es el que produce un conjunto de clasificadores que maximice el área bajo la curva(AUC - Area Under the Curve)(Zulma and Flavio, 2009).



Resultados de la tesis - Señales de Tránsito de Alemania

Tabla: Indicadores de los 5 modelos entrenados para el Dataset - Alemania

Indicadores	Modelo A(%)	Modelo B(%)	Modelo C(%)	Modelo D(%)	Modelo E(%)
PVP	96.14	98.35	98.08	98.37	98.61
PVN	99.93	99.96	99.96	99.96	99.97
PPV	95.69	97.52	97.57	97.78	98.01
AUC-PR	94.31	96.88	96.60	96.93	97.30
AUC-ROC	98.04	99.15	99.02	99.17	99.29
ACC	97.08	98.41	98.27	98.43	98.62



Señales de Tránsito de Alemania

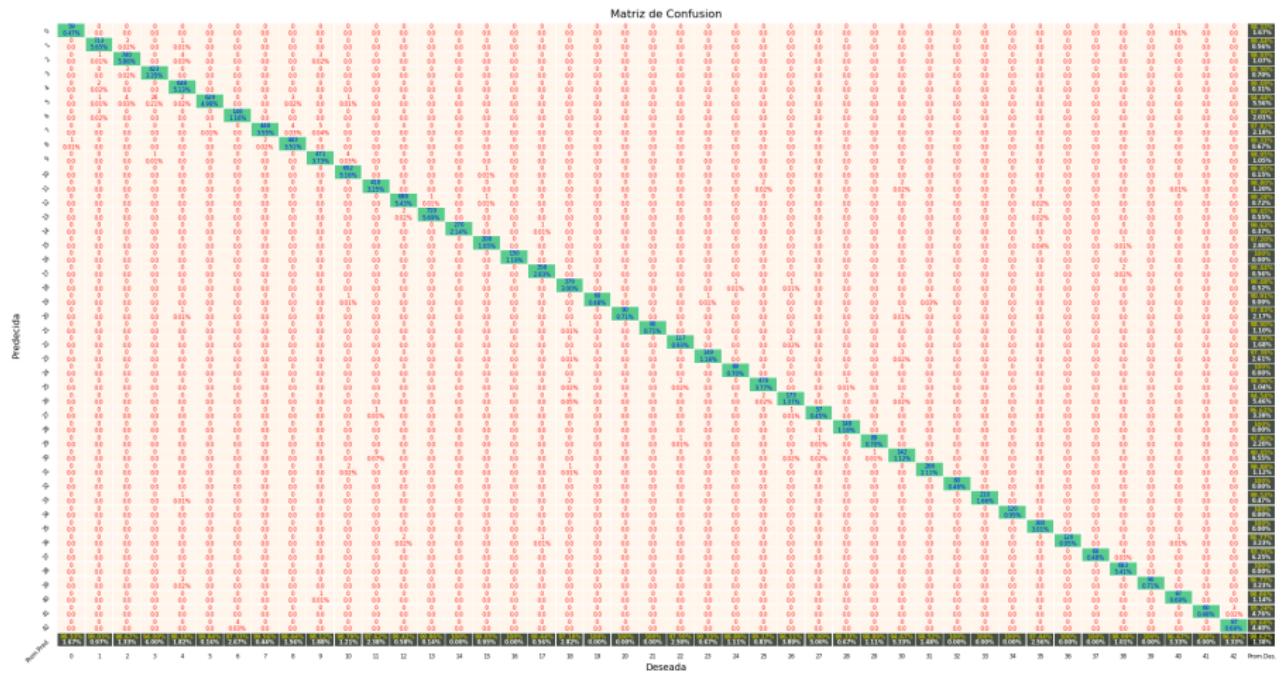


Figura: Matriz de Confusión del Modelo E - Dataset de imágenes de Alemania

Fuente: Elaboración propia



Señales de Tránsito de Alemania

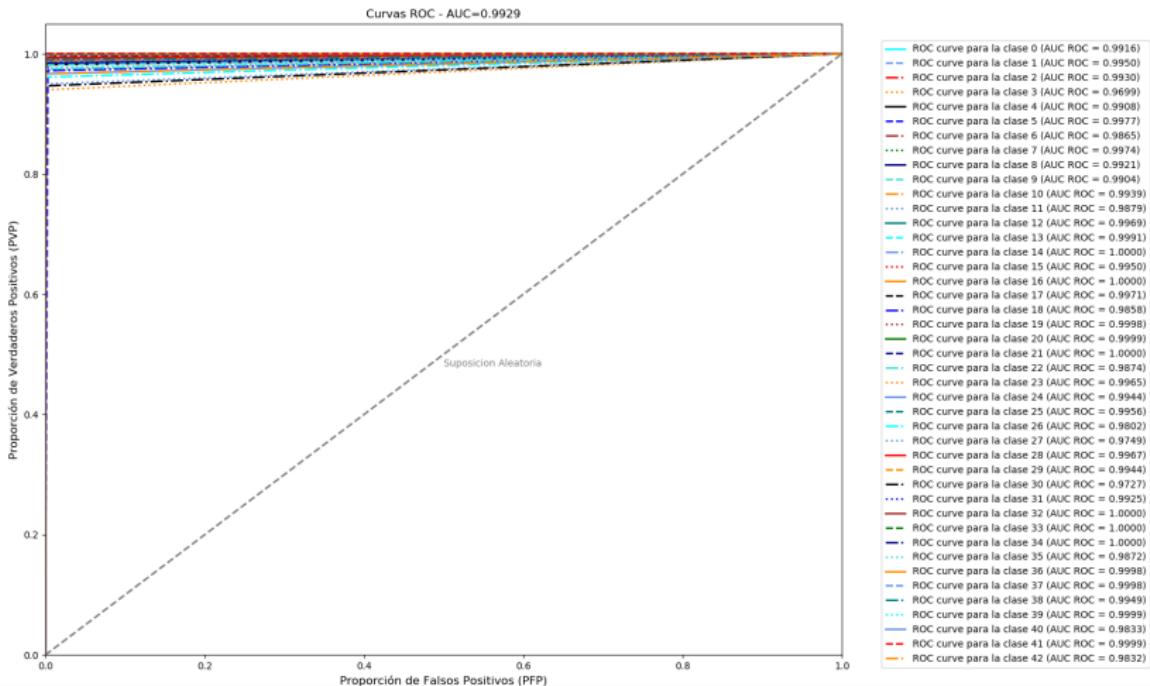


Figura: Área debajo de la Curva ROC del Modelo E - Dataset de imágenes de Alemania

Fuente: Elaboración propia



Señales de Tránsito de Alemania

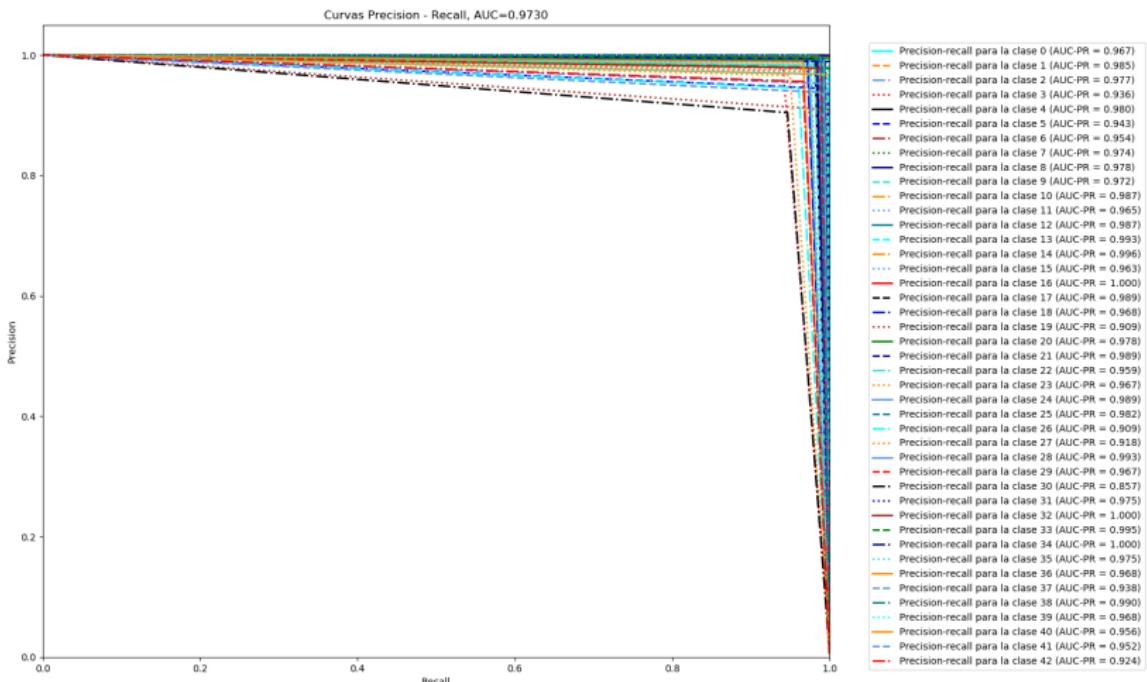


Figura: Área debajo de la Curva PR del Modelo E - Dataset de imágenes de Alemania

Fuente: Elaboración propia



Resultados de la tesis - Señales de Tránsito de Perú

Tabla: Indicadores de los 5 modelos entrenados para el Dataset - Perú

Indicadores	Modelo A(%)	Modelo B(%)	Modelo C(%)	Modelo D(%)	Modelo E(%)
PVP	95.78	97.61	98.21	98.09	98.81
PVN	99.44	99.69	99.75	99.69	99.83
PPV	96.75	98.29	98.51	98.04	98.94
AUC-PR	94.06	96.75	97.33	96.71	98.19
AUC-ROC	97.62	98.66	98.99	98.90	99.33
ACC	96.74	98.23	98.55	98.21	99.02

Fuente: Elaboración propia.



Señales de Tránsito de Perú

		Matriz de Confusión								
		0	1	2	3	4	5	6	Prom. Des.	
Predicida	0	1025 21.82%	1 0.02%	0 0.0	0 0.0	1 0.02%	0 0.0	0 0.0	99.81% 0.19%	
	1	2 0.04%	832 17.71%	3 0.06%	0 0.0	1 0.02%	0 0.0	1 0.02%	99.17% 0.83%	
	2	4 0.09%	4 0.09%	683 14.54%	4 0.09%	1 0.02%	2 0.04%	2 0.04%	97.57% 2.43%	
	3	1 0.02%	5 0.11%	0 0.0	812 17.28%	0 0.0	2 0.04%	2 0.04%	98.78% 1.22%	
	4	1 0.02%	0 0.0	0 0.0	0 0.0	615 13.09%	2 0.04%	1 0.02%	99.35% 0.65%	
	5	0 0.0	0 0.0	1 0.02%	0 0.0	0 0.0	445 9.47%	1 0.02%	99.55% 0.45%	
	6	1 0.02%	0 0.0	0 0.0	1 0.02%	1 0.02%	1 0.02%	240 5.11%	98.36% 1.64%	
		99.13%	98.81%	99.42%	99.39%	99.35%	98.45%	97.17%	99.02%	
		0.87%	1.19%	0.58%	0.61%	0.65%	1.55%	2.83%	0.98%	
		0	1	2	3	4	5	6	Prom. Des.	

Figura: Matriz de Confusión del Modelo E - Dataset de imágenes de Perú

Fuente: Elaboración propia



Señales de Tránsito de Perú

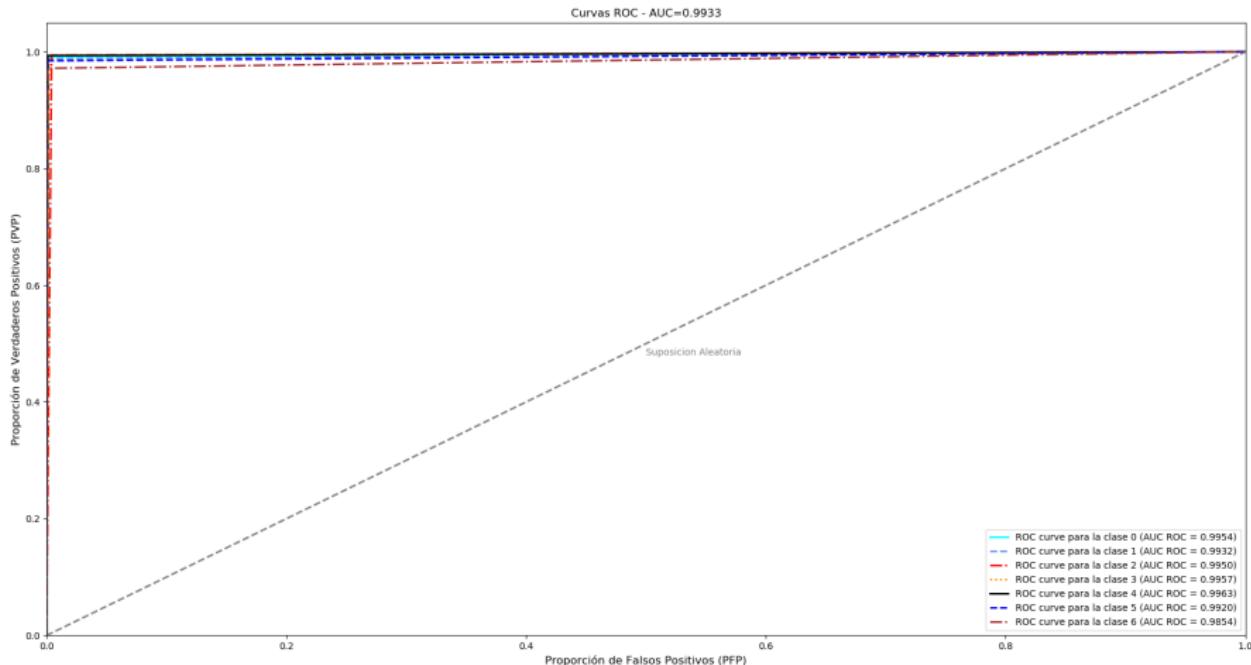


Figura: Área debajo de la Curva ROC del Modelo E - Dataset de imágenes de Perú

Fuente: Elaboración propia



Señales de Tránsito de Perú

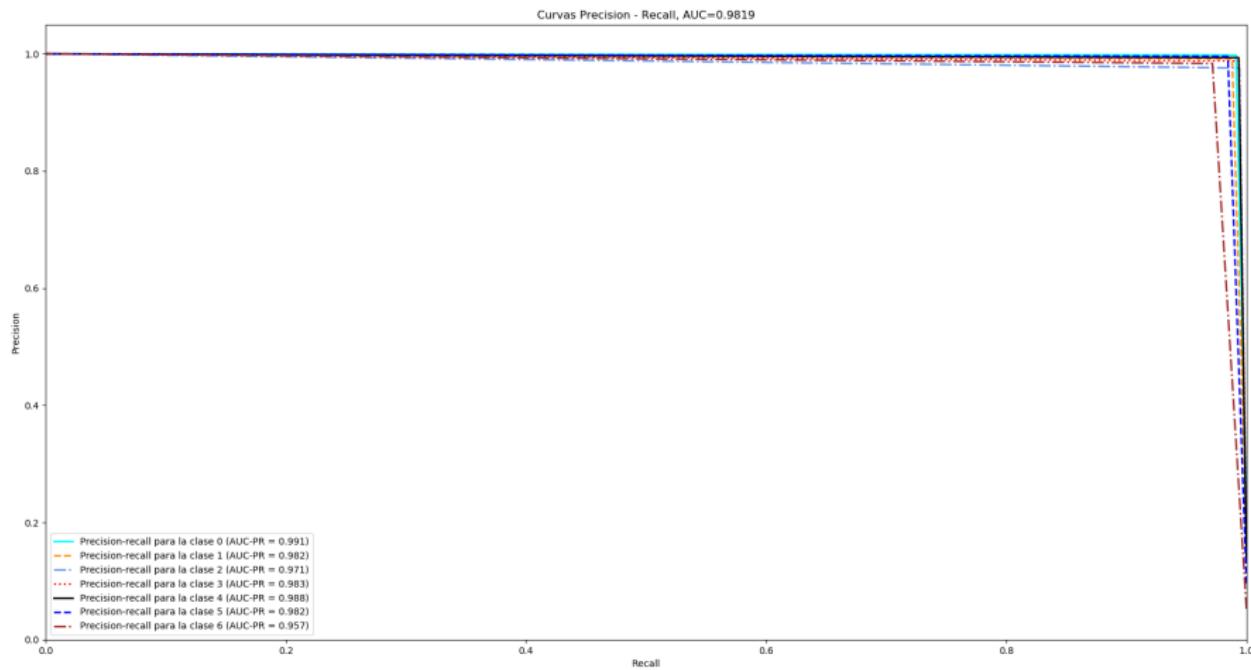


Figura: Área debajo de la Curva PR del Modelo E - Dataset de imágenes de Perú

Fuente: Elaboración propia



Reconocimiento de Señal

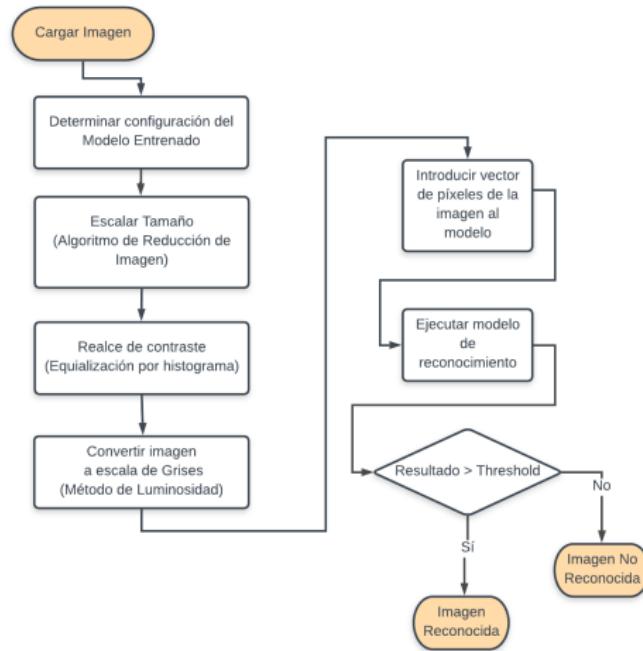


Figura: Flujograma para el reconocimiento de una señal de tránsito



Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

- 1 Introducción
- 2 Motivación
- 3 Formulación del problema e Hipótesis
- 4 Importancia de la investigación
- 5 Contribución de la investigación
- 6 Marco teórico
 - Aprendizaje Profundo
 - Red Convolutacional
 - Arquitectura del Modelo
 - Componentes del Modelo
 - Entrenamiento y Validación
- 7 Propuesta o tema central de la tesis
- 8 Resultados de la tesis
- 9 Consideraciones finales
- 10 Referencias bibliográficas



Consideraciones finales

- **Conclusiones:**



Consideraciones finales

- **Conclusiones:**

- ▶ Se obtuvo dos datasets de imágenes de señales de tránsito, uno de Alemania distribuidas en 43 categorías y otro de Perú distribuidas en 7 categorías, compuestos inicialmente por 51839 imágenes y 614 imágenes respectivamente.



Consideraciones finales

- **Conclusiones:**

- ▶ Se obtuvo dos datasets de imágenes de señales de tránsito, uno de Alemania distribuidas en 43 categorías y otro de Perú distribuidas en 7 categorías, compuestos inicialmente por 51839 imágenes y 614 imágenes respectivamente.
- ▶ Se analizaron los datasets y al dividirlos en 3 grupos (entrenamiento, validación y evaluación), se determinó la necesidad de aumentar ambos y esto fue realizado con la ayuda de técnicas de procesamiento de imágenes.



Consideraciones finales

● Conclusiones:

- ▶ Se obtuvo dos datasets de imágenes de señales de tránsito, uno de Alemania distribuidas en 43 categorías y otro de Perú distribuidas en 7 categorías, compuestos inicialmente por 51839 imágenes y 614 imágenes respectivamente.
- ▶ Se analizaron los datasets y al dividirlos en 3 grupos (entrenamiento, validación y evaluación), se determinó la necesidad de aumentar ambos y esto fue realizado con la ayuda de técnicas de procesamiento de imágenes.
- ▶ Fueron diseñados cinco modelos de red para cada dataset, los cuales fueron entrenados, validados y evaluados individualmente.



Consideraciones finales

● Conclusiones:

- ▶ Se obtuvo dos datasets de imágenes de señales de tránsito, uno de Alemania distribuidas en 43 categorías y otro de Perú distribuidas en 7 categorías, compuestos inicialmente por 51839 imágenes y 614 imágenes respectivamente.
- ▶ Se analizaron los datasets y al dividirlos en 3 grupos (entrenamiento, validación y evaluación), se determinó la necesidad de aumentar ambos y esto fue realizado con la ayuda de técnicas de procesamiento de imágenes.
- ▶ Fueron diseñados cinco modelos de red para cada dataset, los cuales fueron entrenados, validados y evaluados individualmente.
- ▶ Luego de experimentar el uso de diversas funciones de activación, funciones de costo, ajuste de parámetros y métodos de optimización, se logró determinar un conjunto de hiperparámetros de los cuales solo la Tasa de Aprendizaje varía para cada dataset de imágenes.



Consideraciones finales

- **Conclusiones:**



Consideraciones finales

- **Conclusiones:**

- ▶ Al finalizar el análisis de los 10 modelos (cinco por cada dataset), se puede observar que cuanto más profunda sea la red neuronal, se obtienen mejores resultados.



Consideraciones finales

● Conclusiones:

- ▶ Al finalizar el análisis de los 10 modelos (cinco por cada dataset), se puede observar que cuanto más profunda sea la red neuronal, se obtienen mejores resultados.
- ▶ El modelo final obtenido compuesto principalmente de 4 capas convolucionales, 2 capas totalmente conectadas y un total de aproximadamente 2 millones de neuronas, contribuye en el reconocimiento de señales de Tránsito de Alemania con una tasa de acierto del **98.62 %**, mucho mejor que el resultado de 95.29 % obtenido por Ayuque Arenas (2016) y mucho más próximo al mejor resultado de 99.46 % obtenido en las investigaciones hechas por Cireşan et al. (2012) en base al dataset GTSRB.



Consideraciones finales

● Conclusiones:

- ▶ Al finalizar el análisis de los 10 modelos (cinco por cada dataset), se puede observar que cuanto más profunda sea la red neuronal, se obtienen mejores resultados.
- ▶ El modelo final obtenido compuesto principalmente de 4 capas convolucionales, 2 capas totalmente conectadas y un total de aproximadamente 2 millones de neuronas, contribuye en el reconocimiento de señales de Tránsito de Alemania con una tasa de acierto del **98.62 %**, mucho mejor que el resultado de 95.29 % obtenido por Ayuque Arenas (2016) y mucho más próximo al mejor resultado de 99.46 % obtenido en las investigaciones hechas por Cireşan et al. (2012) en base al dataset GTSRB.
- ▶ Para el dataset de señales de tránsito vehicular del Perú, el modelo E con el mismo diseño neuronal y similares configuraciones compuesto por aproximadamente 6 millones de neuronas permite también obtener un **alto grado de acierto (99.02 %)** tras analizar 4698 imágenes.



Consideraciones finales

- En conclusión, el objetivo general que trata sobre implementar un modelo basado en el aprendizaje profundo de redes neuronales convolucionales para reconocer automáticamente señales de tránsito vehicular fue conseguido a través del Modelo E.



Consideraciones finales

- Trabajos futuros:



Consideraciones finales

- **Trabajos futuros:**

- ▶ El modelo puede ser mejorado(ampliado), teniendo muchas más capas convolucionales y capas totalmente conectadas para poder experimentar si existe o no alguna mejora en los resultados. Además, se recomienda obtener muchas más imágenes para exceder el rendimiento humano, (Goodfellow et al., 2016)



Consideraciones finales

- **Trabajos futuros:**

- ▶ El modelo puede ser mejorado(ampliado), teniendo muchas más capas convolucionales y capas totalmente conectadas para poder experimentar si existe o no alguna mejora en los resultados. Además, se recomienda obtener muchas más imágenes para exceder el rendimiento humano, (Goodfellow et al., 2016)
- ▶ El dataset de señales de tránsito del Perú también puede ser ampliado con la finalidad de abarcar más categorías, ya que se tiene confianza por lo mostrado con el dataset de Alemania que el modelo es robusto para soportar mayor cantidad de estas.



Consideraciones finales

- **Trabajos futuros:**

- ▶ El modelo puede ser mejorado(ampliado), teniendo muchas más capas convolucionales y capas totalmente conectadas para poder experimentar si existe o no alguna mejora en los resultados. Además, se recomienda obtener muchas más imágenes para exceder el rendimiento humano, (Goodfellow et al., 2016)
- ▶ El dataset de señales de tránsito del Perú también puede ser ampliado con la finalidad de abarcar más categorías, ya que se tiene confianza por lo mostrado con el dataset de Alemania que el modelo es robusto para soportar mayor cantidad de estas.
- ▶ Se sugiere integrar el modelo obtenido en un sistema más general que primero localice las señales de tránsito en escenas que abarcan más de una señal de tránsito, para luego proceder a su multireconocimiento.



Modelo De Reconocimiento Automático De Señales De Tránsito Vehicular Mediante Aprendizaje Profundo De Redes Neuronales Convolucionales

- 1 Introducción
- 2 Motivación
- 3 Formulación del problema e Hipótesis
- 4 Importancia de la investigación
- 5 Contribución de la investigación
- 6 Marco teórico
 - Aprendizaje Profundo
 - Red Convolutacional
 - Arquitectura del Modelo
 - Componentes del Modelo
 - Entrenamiento y Validación
- 7 Propuesta o tema central de la tesis
- 8 Resultados de la tesis
- 9 Consideraciones finales
- 10 Referencias bibliograficas



Referencias bibliograficas

- Ayuque Arenas, K. J. M. (2016). Diseño de un sistema de clasificación de señales de tránsito vehicular utilizando redes neuronales convolucionales. *Universidad San Ignacio de Loyola*.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.
- Cireşan, D., Meier, U., Masci, J., and Schmidhuber, J. (2012). Multi-column deep neural network for traffic sign classification. *Neural Networks from IJCNN 2011*, 32(C):333 – 338. <http://www.deeplearningbook.org>.
- Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves.
- Dong, X. and Zhou, D.-X. (2008). Learning gradients by a gradient descent algorithm. *Journal of Mathematical Analysis and Applications*, 341(2):1018 – 1027.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press



Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., and Wang, G. (2015). Recent advances in convolutional neural networks. *CoRR*, abs/1512.07108.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA. ACM.

Jordan, J. (2018). Setting the learning rate of your neural network. <https://www.jeremyjordan.me/nn-learning-rate>. [Online; accesado Noviembre 23, 2018].

Karpathy, A. (2016). Convolutional neural networks (cnn / convnets). <http://cs231n.github.io/convolutional-networks/>. [Online; accesado Noviembre 23, 2018].

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). *ImageNet Classification with Deep Convolutional Neural Networks*. Curran Associates, Inc.

LeCun, Y., L.Bottou, Y.Bengio, and P.Haffner (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86.



- Moore, A. W. (2001). Cross-validation for detecting and preventing overfitting.
School of Computer Science Carnegie Mellon University.
- Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA. Omnipress.
- OMS (2017). Organización mundial de la salud - 10 datos sobre la seguridad vial en el mundo. <http://www.who.int/features/factfiles/roadsafety/es/>. [Online; accesado Agosto 23, 2018].
- Rohrer, B. (2016). How convolutional neural networks work. [Online; accesado Octubre 23, 2018].
- Romero, R. F. (2015a). Scc0270/scc5809 - redes neurais - deep learning. *ICMC - USP*.
- Romero, R. F. (2015b). Scc0270/scc5809 - redes neurais - multilayer perceptron. *ICMC - USP*.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going deeper with convolutions. *CoRR*, abs/1409.4842.



- Yadav, G., Maheshwari, S., and Agarwal, A. (2014). Contrast limited adaptive histogram equalization based enhancement for real time video system. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2392–2397.
- Zulma, S. and Flavio, P. (2009). Procesamiento de imágenes para la clasificación de café cerez. *PROSPECTIVA*, 7:67–73.

