CNN para reconocimiento del alfabeto ASL en imágenes

Daniel Queijeiro Albo A01710441

ABSTRACT. El propósito de este documento es presentar todo el proceso para entrenar una red neuronal convolucional (CNN) para clasificar señas del alfabeto ASL (American Sign Language) a partir de imágenes RGB. El conjunto de datos proviene de Kaggle. El objetivo es predecir la letra correspondiente a cada imagen con alta exactitud (70%) haciendo uso de la librería Tensorflow.

1. INTRODUCCIÓN

El lenguaje de señas americano (ASL, por sus siglas en inglés) es un sistema de comunicación visual fundamental para la comunidad sorda. El desarrollo de sistemas automáticos de reconocimiento de señas puede facilitar la comunicación y reducir las barreras de accesibilidad.

Se utilizó el dataset ASL Alphabet disponible en Kaggle. Este conjunto de datos contiene imágenes de 200 x 200 píxeles en formato RGB, representando diferentes configuraciones de manos correspondientes a cada símbolo del alfabeto ASL.

---

2. DESCRIPCIÓN DEL DATASET

El dataset está conformado por 87,000 imágenes correspondientes a 29 clases, de las cuales 26 de ellas son para las letras A - Z y las otras 3 para “Space”, “Delete” y “Nothing”.

La inclusión de las 3 clases extra es un aspecto crítico del dataset. En una aplicación en tiempo real (ej. un traductor de video en vivo), el modelo no solo debe reconocer las letras, sino también:

- Nothing: Identificar cuándo el usuario no está haciendo ninguna seña (ej. la mano está en reposo). Esto es vital para evitar una avalancha de predicciones incorrectas (falsos positivos).
- Space: Reconocer la seña de "espacio", permitiendo al sistema delimitar palabras y frases de forma coherente.
- Delete: Entender la seña para "borrar" o "corregir", lo cual es crucial para construir una interfaz de usuario interactiva y funcional.

3. ETL

Debido al origen del dataset, no fue necesario hacer una limpieza exhaustiva al dataset. Lo principal que se alteró del dataset fue crear una separación de los datos de entrenamiento para crear datos de validación, siguiendo el patrón de 80% para entrenamiento y 20% para validación. El dataset igualmente ya incluye datos para prueba, pero para este estudio se promueve el uso de imágenes propias para las pruebas.

El data augmentation es una técnica que crea variaciones artificiales de las imágenes de entrenamiento mediante transformaciones geométricas y

fotométricas. Esto aumenta la diversidad del dataset y mejora la capacidad de generalización del modelo.

En nuestro caso queremos que el modelo aprenda a identificar las señas aún con variaciones en pose, centrado o iluminación, por lo que usando un pipeline de data augmentation realizamos los siguientes cambios (únicamente al conjunto de entrenamiento)

1. RandomFlip("horizontal")
2. RandomRotation(0.1)
3. RandomZoom(0.2)
4. RandomTranslation(0.2, 0.2)

---

4. CONSTRUCCIÓN DEL MODELO

4a. CNNs

Una CNN es un tipo de red neuronal artificial que imita la forma en que el humano procesa la información visual. Son el estándar de la industria para tareas de visión por computadora. Sus componentes clave incluyen:

- Capas Convolucionales (Conv2D): Actúan como detectores de características. Aplican filtros a la imagen de entrada para identificar patrones, como bordes, texturas o formas. En las capas más profundas, estos filtros aprenden a reconocer características más complejas.
- Capas de Agrupación (MaxPooling2D): Reducen las dimensiones espaciales (ancho y alto) de la imagen. Esto hace que el modelo sea más eficiente y ayuda a que la red sea robusta a pequeñas variaciones en la posición de la seña dentro de la imagen.
- Capas Densas (Dense): Son las capas de una red neuronal tradicional. Se colocan al final de la CNN para tomar las características de alto nivel detectadas por las capas convolucionales y realizar la clasificación final.

4b. Arquitectura del modelo

Se diseñó una arquitectura CNN secuencial que acepta imagenes de tamaño 200 x 200 x 3 (ancho, alto y canales de color). Dicha arquitectura se compone de cuatro bloques convolucionales principales, seguidos de un bloque de clasificación.

En los bloques convolucionales se van aumentando los filtros progresivamente (32, 64, 128 y 256) para que el modelo pueda aprender características desde muy simples hasta muy complejas.

Y en el bloque de clasificación obtenemos al final nuestra probabilidad de que la imagen pertenezca a cada una de las 29 imágenes o mejor dicho, su predicción.

Además, se implementaron 3 configuraciones para mejorar la gestión del entrenamiento. Se agregó un “Early stopping” para que se detenga el entrenamiento automáticamente si la pérdida no mejora después de 10 épocas. También se reduce el learning rate a la mitad si es que la pérdida se estanca durante 5 épocas. Y finalmente, se guarda la mejor versión del modelo (basado en la pérdida) por si el modelo fluctúa en su accuracy.

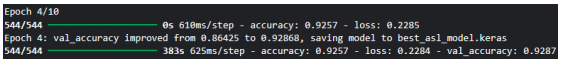
---

5. RESULTADOS

5a. Métricas

El modelo se entrenó utilizando la configuración descrita anteriormente. El rendimiento se evaluó en los conjuntos de datos en entrenamiento, validación y prueba.

El modelo mostró una curva de aprendizaje estable, con el accuracy aumentando y el loss disminuyendo consistentemente



5b. Conclusiones

Con los resultados obtenidos es evidente que el modelo CNN diseñado logra capturar de manera robusta la relación entre las imágenes de señas y las 29 clases del alfabeto ASL.

Sin embargo, el hecho de que aun exista ese margen de error nos indica que hay una fracción de varianza que el modelo no loga explicar por completo. Esto probablemente se deba a que ciertas clases poseen una alta similitud en su forma de señalizar, o a que existen interacciones más complejas, como variaciones sutiles en iluminación, ángulo de la mano o fondo, que no fueron completamente generalizadas a pesar del data augmentation.