Detección de sonrisas con deep learning y edge computing

Jorge Guijarro[1], Frado García[2], Claudio Gonzalez[3] and Gabriel Meléndez[4]

[1, 2]Tecnológico de Monterrey, Campus Guadalajara

November 29, 2024

Abstract

Este reporte analiza y compara dos enfoques diferentes basados en redes neuronales para clasificar imágenes de rostros en dos categorías: sonriendo y no sonriendo. El primer enfoque utiliza una red neuronal convolucional (CNN), mientras que el segundo implementa un autoencoder configurado como "detector de anomalías", entrenado únicamente con imágenes de rostros sonrientes. Ambos modelos se evalúan en términos de precisión, desempeño general y tiempo de ejecución, destacando las ventajas y limitaciones de cada método. Este trabajo tiene como objetivo determinar cuál de los dos enfoques es más adecuado para resolver este problema en escenarios prácticos.

Palabras Claves: CNN, Autoencoder, Accuracy, imágenes

1 Introducción

En la actualidad, la clasificación automática de imágenes ha adquirido una gran relevancia debido a su amplia gama de aplicaciones prácticas, que incluyen desde sistemas de seguridad hasta interfaces más intuitivas en dispositivos personales. Dentro de este campo, la detección y clasificación de rostros sonrientes versus no sonrientes plantea un desafío único y significativo, dado que los gestos faciales son indicadores clave de las emociones humanas.

Este trabajo se centra en abordar este problema utilizando dos métodos complementarios basados en aprendizaje automático. Por un lado, una red neuronal convolucional (CNN) se entrena de forma tradicional para clasificar imágenes en las categorías de "sonriente" y "no sonriente". Por otro lado, se emplea un autoencoder como detector de anomalías, un enfoque menos convencional que permite entrenar al modelo únicamente con imágenes de una clase (rostros sonrientes) y evaluar su capacidad para identificar las imágenes que no pertenecen a esta categoría.

Ambos modelos son entrenados y evaluados utilizando el mismo conjunto de datos, lo que asegura una comparación justa en términos de desempeño y tiempo de ejecución. Adicionalmente, este proyecto explora las implicaciones prácticas de ambos enfoques, considerando sus aplicaciones potenciales en sistemas en tiempo real utilizando Edge Computing, donde los recursos de cómputo suelen ser limitados.

Este documento presenta una descripción detallada de los modelos, su metodología de entrenamiento y evaluación, y un análisis comparativo de los resultados obtenidos.

2 METODOLOGÍA

2.1 Descripción del Dataset

El dataset utilizado consta de dos partes principales: la primera parte es una carpeta llamada ”faces”, que contiene todas las imágenes de los rostros, tanto con sonrisa como sin sonrisa. Estas imágenes están en formato .ppm. La segunda parte consiste en dos archivos .txt: el primero, ”NON-SMILE_list.txt”, contiene la lista de las imágenes de rostros sin sonrisa, y el segundo, ”SMILE_list.txt”, incluye las imágenes de rostros con sonrisa. Estos archivos sirven como etiquetas (labels) que utilizaremos para nuestro conjunto de entrenamiento.

2.2 Preprocesamiento de los Datos

2.2.1 Conversión de formato

Dado que las imágenes originales estaban en formato .ppm, fue necesario convertirlas al formato .jpg para garantizar que los modelos no tuvieran problemas en ser entrenados con las herramientas y librerías como Tensorflow, además, los archivos de texto ”NON-SMILE_list.txt” y ”SMILE_list.txt” contenían las etiquetas de las imágenes en formato .jpg. Esto se logró mediante un script que iteró sobre las imágenes en el directorio, utilizando la librería PIL para realizar la conversión y guardar los resultados en un nuevo directorio.

2.2.2 Organización del Dataset

A partir de los archivos NON-SMILE_list.txt y SMILE_list.txt, las imágenes se organizaron en carpetas separadas según su clase (smiling y not_smiling). De esta manera conseguimos una mejor organización de los datos, que nos es útil principalmente para el modelo Autoencoder que es entrenado únicamente con una clase.

2.2.3 Normalización de las imágenes

Todas las imágenes se redimensionaron a 64x64 píxeles y se escalaron a valores entre 0 y 1 dividiendo los valores de los píxeles entre 255. De esta manera logramos que nuestros modelos puedan converger más rápidamente durante el entrenamiento.

2.3 Arquitectura de la CNN

La primera red implementada fue una red neuronal convolucional (CNN) cuyo diseño abordó la tarea de clasificación binaria entre rostros sonrientes y no sonrientes.

La red consta de tres bloques convolucionales en la que se aplica un kernel de 3x3 con 32, 64 y 128 filtros respectivamente y utilizando una función de actuvación ReLU en cada una. Cada una de estas capas se complementa con una operación MaxPooling de 2x2 para reducir la dimensión y conservar las características más importantes.
Luego se utiliza una capa densa de 128 unidades y función de activación ReLU que es complementada con una operación MaxPooling para finalmente pasar a la capa de salida que implementa una función de activación sigmoid que arroja probabilidades entre 0 y 1 para clasificar si la imágen se trata de un rostro con una sonrisa o un rostro sin sonrisa.

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d_5 (MaxPooling2D)	(None, 31, 31, 32)	0
conv2d_7 (Conv2D)	(None, 29, 29, 64)	18,496
max_pooling2d_6 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_8 (Conv2D)	(None, 12, 12, 128)	73,856
max_pooling2d_7 (MaxPooling2D)	(None, 6, 6, 128)	0
flatten_1 (Flatten)	(None, 4608)	0
dense_2 (Dense)	(None, 128)	589,952
dropout_5 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 1)	129

Figure 1. Arquitectura del modelo CNN

2.4 Arquitectura Autoencoder

El segundo enfoque que se utilizó fue un autoencoder que se implementó como un modelo no supervisado enfocado en la reconstrucción de imágenes para la detección de anomalías. Este modelo consta de dos principales bloques:

2.4.1 Encoder

Primero Utiliza dos capas convolucionales con 32 y 64 kernels de (3x3) respectivamente, cada uno seguido de una operación de max-pooling y un dropout final. De esta forma representamos las imágenes y capturamos las características más relevantes en una dimensión menor.

2.4.2 Decoder

El Decoder reconstruye las imágenes mediante 2 capas Conv2DTranspose utilizando 32 y 64 kernels de (3x3) para la primera y segunda capa respectivamente, con funciones de activación ReLU y seguidas de operaciones Dropout. Finalmente, la última capa genera imágenes reconstruidas con los mismos valores que las originales (entre 0 y 1), utilizando una una función de activación sigmoid

3 EXPERIMENTOS

3.1 División de los Datos

Los datos se dividieron en conjuntos de entrenamiento, validación y prueba con proporciones dle 64%, 16% y 20%, respectivamente.

Para el caso del Autoencoder se utilizó el mismo conjunto de entrenamiento pero solo se conservó la clase ”sonriente” (smiling).

Layer (type)	Output Shape	Param #
input_layer_3 (InputLayer)	(None, 64, 64, 3)	0
conv2d_9 (Conv2D)	(None, 64, 64, 32)	896
max_pooling2d_8 (MaxPooling2D)	(None, 32, 32, 32)	0
dropout_6 (Dropout)	(None, 32, 32, 32)	0
conv2d_10 (Conv2D)	(None, 32, 32, 64)	18,496
max_pooling2d_9 (MaxPooling2D)	(None, 16, 16, 64)	0
dropout_7 (Dropout)	(None, 16, 16, 64)	0
conv2d_11 (Conv2D)	(None, 16, 16, 64)	36,928
dropout_8 (Dropout)	(None, 16, 16, 64)	0
conv2d_transpose_2 (Conv2DTranspose)	(None, 32, 32, 32)	18,464
dropout_9 (Dropout)	(None, 32, 32, 32)	0
conv2d_transpose_3 (Conv2DTranspose)	(None, 64, 64, 3)	867

Figure 2. Arquitectura del modelo autoencoder

3.2 Parámetos de entrenamiento del modelo CNN

- Optimizador: Adam
- Learning rate: 0.001
- Función de pérdida: Entropía cruzada binaria
- Métrica: Exactitud
- Número de épocas: 30

3.3 Parámetos de entrenamiento del modelo Autoencoder

- Optimizador: Adam
- Learning rate: 0.001
- Función de pérdida: MSE
- Número de épocas: 30

4 Resultados

4.1 Resultados obtenidos con el modelo CNN

La CNN mostró un desempeño sobresaliente, alcanzando una precisión global del 95% en el conjunto de prueba y un F1-score promedio de 0.96. Estos resultados reflejan su efectividad para clasificar

imágenes de ambas categorías. La matriz de confusión confirmó una alta precisión en ambas clases, con tasas mínimas de falsos positivos y falsos negativos. En general, la CNN demostró ser un modelo robusto y adecuado para la tarea de clasificación binaria, destacándose principalemente por su enfoque supervisado.

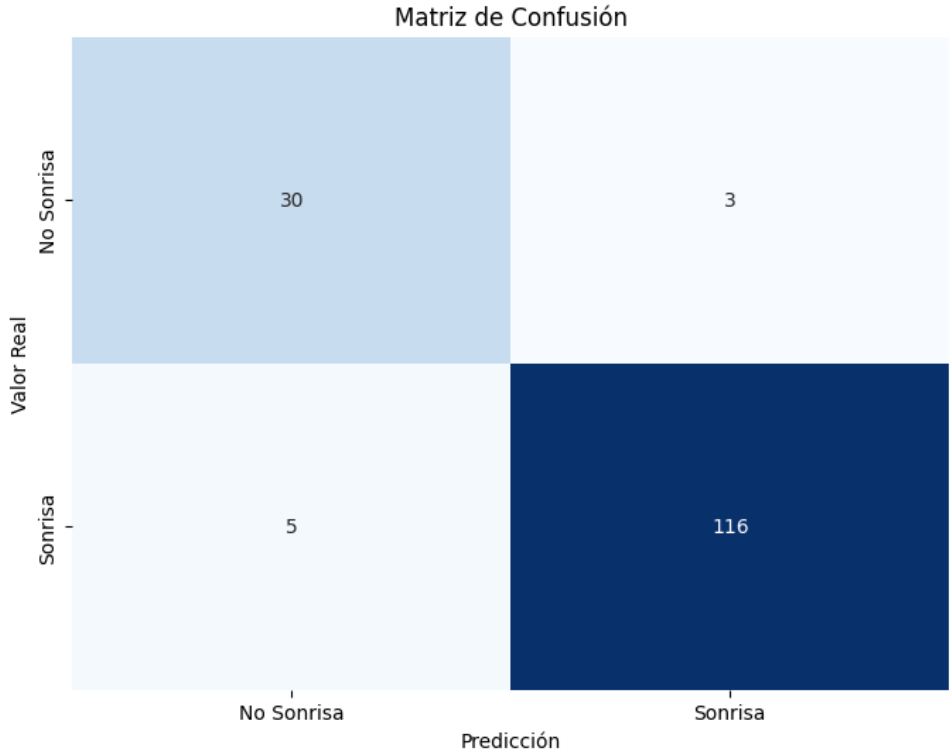

Figure 3. Matriz de confusión del modelo CNN

4.2 Resultados obtenidos con el modelo Autoencoder

El autoencoder, por su parte, presentó un desempeño considerablemente más limitado, con una precisión global del 46% y un F1-score promedio de 0.45. La matriz de confusión evidenció altas tasas de falsos positivos y falsos negativos, lo que indicó dificultades para discriminar entre las clases. Este desempeño refleja las limitaciones de utilizar un modelo basado en detección de anomalías y entrenado exclusivamente con imágenes ”sonrientes” para una tarea de clasificación binaria. Sería interesante probar si al usarse un mayor número de observaciones (fotografías) pueda mejorar considerablemente su rendimiento a la hora de detectar anomalías (caras no sonrientes).

5 Conclusión

Los resultados obtenidos muestran que la Red Neuronal Convolucional (CNN) es significativamente superior al autoencoder para la tarea de clasificación de imágenes de sonrisas. La CNN no solo presentó métricas de rendimiento mucho más altas, sino que también demostró una mayor capacidad para generalizar a nuevas imágenes, lo que la convierte en la opción preferida para este tipo de problemas. Diferencias Clave entre la CNN y el Autoencoder:

1. Arquitectura y propósito:

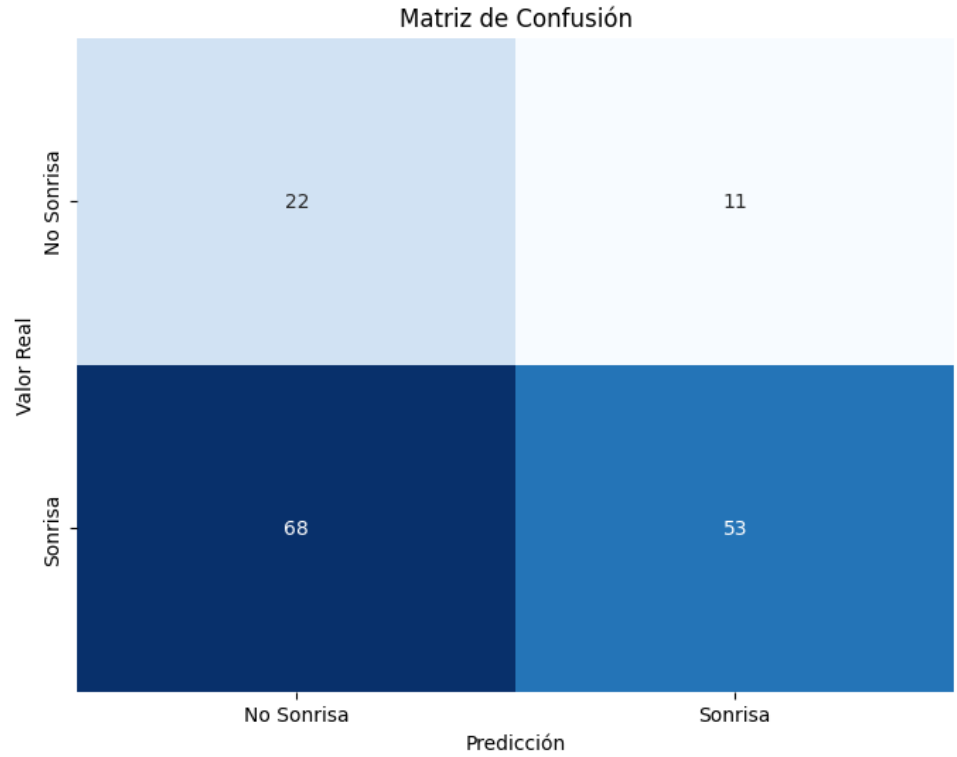

Figure 4. Matriz de confusión del modelo Autoencoder

- **CNN**: Está diseñada específicamente para trabajar con datos de imágenes, aprovechando las convoluciones para extraer características espaciales importantes de las imágenes. Esta arquitectura le permite aprender patrones complejos y realizar tareas de clasificación con alta precisión.
- **Autoencoder**: En su forma tradicional, es un modelo no supervisado que se utiliza para la reducción de dimensionalidad y la detección de anomalías, no para clasificación directa. Aunque se puede adaptar para clasificación, no está optimizado para esta tarea, lo que limita su rendimiento en comparación con la CNN.

2. Capacidad para capturar características relevantes:

   - **CNN**: A través de sus capas convolucionales, la CNN es capaz de identificar y aprender características importantes de las imágenes, como bordes, texturas y formas que son esenciales para identificar una sonrisa en una imagen.
   - **Autoencoder**: Si bien puede aprender representaciones compactas de las imágenes, no está diseñado para capturar características específicas relacionadas con las clases de interés (como la presencia de una sonrisa), lo que disminuye su rendimiento en clasificación.

3. Adaptabilidad al tipo de datos:

   - **CNN**: Es altamente efectiva para problemas de clasificación de imágenes, ya que está específicamente entrenada para reconocer patrones espaciales y relaciones locales en los datos visuales.

- **Autoencoder**: Aunque puede ser útil en problemas de detección de anomalías y reducción de dimensionalidad, no es tan efectivo cuando se requiere una clasificación precisa de categorías bien definidas, como la clasificación de sonrisas en imágenes.

4. Métricas de desempeño:

   - **CNN**: Los resultados de la CNN fueron significativamente mejores en casi todas las métricas clave, incluidos la precisión, el recall, el F1-score y la exactitud. Este modelo mostró una capacidad mucho mayor para distinguir entre imágenes con y sin sonrisa.
   - **Autoencoder**: Aunque el autoencoder presentó un desempeño aceptable en términos de recall para la clase sin sonrisa, su bajo rendimiento en la clasificación de sonrisas lo hizo menos adecuado para la tarea.

La CNN demostró ser el enfoque más adecuado para la tarea de clasificación de imágenes de sonrisas debido a su capacidad para aprender y generalizar patrones espaciales en imágenes, lo que le permite obtener un rendimiento sobresaliente en precisión y recall. Por el contrario, el autoencoder, aunque útil en otros contextos, mostró ser ineficaz para la clasificación directa de imágenes en este caso específico. Estos resultados sugieren que, para tareas de clasificación de imágenes, especialmente aquellas que requieren un alto grado de precisión, las redes neuronales convolucionales son una opción mucho más potente y adecuada que los autoencoders.