Guatemala, 15 de noviembre 2023

Computer Vision

**Computer Vision – Proyecto Final**

**Bird Detector**

Marcela Melgar

20200487

**Introducción**

Este reporte presenta un análisis detallado y los resultados obtenidos de un modelo de Red Neuronal Convolucional (CNN), diseñado y entrenado para la clasificación de especies de aves.

El modelo fue desarrollado utilizando un conjunto de datos compuesto por imágenes categorizadas de 100 especies de aves, obtenidas del directorio **/kaggle/input/100-bird-species**. Este conjunto se divide en datos de entrenamiento y validación, permitiendo así evaluar la efectividad y precisión del modelo en condiciones controladas.

La metodología empleada incluye el uso de técnicas de aumento de datos y normalización para mejorar la generalización del modelo. La arquitectura de la red se compone de varias capas convolucionales, capas de normalización por lotes, y capas densas, configuradas para optimizar la clasificación multi-clase. Además, se implementaron estrategias como el uso de un optimizador Adam y la aplicación de la técnica de detención temprana para prevenir el sobreajuste.

Este reporte busca no solo documentar el proceso de diseño y entrenamiento del modelo, sino también discutir los resultados obtenidos, proporcionar una valoración crítica de su desempeño y explorar posibles aplicaciones y mejoras futuras.

**Data**

En total, el dataset comprende 84,635 imágenes para entrenamiento, 2,625 imágenes para pruebas y 2,625 para validación, asignando 5 imágenes por especie en los conjuntos de prueba y validación. Cada imagen, de alta calidad, muestra un solo pájaro que ocupa al menos el 50% de los píxeles, lo cual es beneficioso para el entrenamiento y prueba de modelos. Todas las imágenes son originales, sin haber sido creadas por técnicas de aumento de datos. El dataset actualizado contiene 525 especies de aves.

Las imágenes son a color, con un formato de 224 x 224 x 3 en JPG. El dataset incluye un conjunto de entrenamiento, prueba y validación, cada uno dividido en 525 subdirectorios, uno por cada especie

**Métodos**

El proyecto empleó una CNN para la tarea de clasificación de especies de aves. La CNN es ideal para el procesamiento de imágenes debido a su capacidad para detectar patrones jerárquicos y características locales en las imágenes. En la fase de preprocesamiento, se utilizó **ImageDataGenerator** para normalizar los valores de los píxeles y aplicar técnicas de aumento de datos como cizallamiento, zoom y volteo horizontal. Estas transformaciones ayudan a mejorar la generalización del modelo al introducir variabilidad en el conjunto de entrenamiento.

El modelo CNN se definió con múltiples capas, incluyendo capas convolucionales para la extracción de características, capas de normalización por lotes para estabilizar el aprendizaje, y capas densas para la clasificación. Se optó por el optimizador Adam con una tasa de aprendizaje de 0.001, considerado eficaz para una amplia gama de problemas de aprendizaje profundo.

Las capas utilizadas en el modelo son:

* Conv2D
  + Conv2D se refiere a una capa convolucional 2D, que es una de las piezas fundamentales en una red neuronal convolucional, especialmente utilizada en el procesamiento de imágenes. Esta capa aplica un número especificado de filtros convolucionales a la imagen de entrada, en este caso fueron 32 y 64. Cada filtro extrae características específicas (como bordes, texturas, etc.) de la imagen. La "2D" indica que los filtros se aplican en dos dimensiones (altura y anchura de la imagen).
* BatchNormalization()
  + BatchNormalization es una técnica para mejorar la velocidad, el rendimiento y la estabilidad de las redes neuronales artificiales. Normaliza las activaciones de la capa anterior por lote, es decir, ajusta y escala las activaciones para que tengan una media cercana a 0 y una desviación estándar cercana a 1. Esto ayuda a reducir el problema conocido como "desplazamiento de la covarianza" y a menudo permite usar tasas de aprendizaje más altas, acelerando el entrenamiento.
* MaxPooling2D((2, 2))
  + MaxPooling2D es una capa que se utiliza para reducir la dimensionalidad espacial (altura y anchura) de la entrada. Esto se hace tomando el valor máximo de una ventana (en este caso de tamaño 2x2) y desplazándose por la imagen. El pooling ayuda a reducir el número de parámetros y la cantidad de cómputo en la red, controlando así el sobreajuste.
* Flatten()
  + Flatten es una operación que convierte los datos multidimensionales de la capa anterior en un vector 1D. Esta capa no tiene parámetros; simplemente cambia la forma de los datos para que puedan ser introducidos en una capa densa.
* Dense(256) y Dense(500)
  + Las capas Dense son capas de redes neuronales completamente conectadas. Dense(256) crea una capa con 256 neuronas. Cada neurona en una capa densa recibe entradas de todas las neuronas de la capa anterior, es decir, están "densamente" conectadas. Estas capas se utilizan generalmente para aprender patrones no lineales. Dense(500) sigue el mismo principio pero con 500 neuronas. Estas capas suelen usarse para la clasificación o regresión al final de la red neuronal.
* Adam()
  + Adam es un algoritmo de optimización utilizado para actualizar los pesos de la red durante el entrenamiento. Es una mejora de los métodos de descenso de gradiente estocástico (SGD). Adam es popular porque generalmente requiere menos ajuste de hiperparámetros y es bastante eficiente en términos de cálculo.

Cada uno de estos componentes juega un papel vital en el rendimiento y la eficacia de un modelo de CNN, ayudando en la extracción de características, normalización, reducción de la complejidad y clasificación final.

El entrenamiento se realizó utilizando model.fit con los generadores de datos de entrenamiento y validación,. Se utilizó un batch size de 64, con 20 epochs y 1322 steps per epoch. Se redimensionaron las imágenes a 28x28 para un procesamiento con mayor velocidad.

La evaluación del modelo se centró primordialmente en la métrica de Accuracy, que es esencial en la validación de modelos de clasificación como nuestra Red Neuronal Convolucional (CNN) para la identificación de especies de aves. El Accuracy se calcula como la proporción de predicciones correctas frente al total de predicciones hechas por el modelo. Este indicador es particularmente valioso ya que proporciona una comprensión clara y directa de qué tan bien el modelo puede identificar correctamente las diferentes especies de aves en el conjunto de datos. Un valor alto de Accuracy indica que el modelo ha logrado aprender y generalizar eficazmente a partir de los datos de entrenamiento, demostrando así su eficacia en clasificar con precisión nuevas imágenes en los conjuntos de validación y prueba.

Dentro de las posibles mejoras que se pueden hacer, es que el modelo muestra un 56% de accuracy. Esto, a pesar de ser el accuracy más alto que se logró pues este empezó a ser de 0.12%, luego 15%, 51% hasta 56%. Puede mejorarse este accuracy y permitir que el modelo sea más efectivo para poder clasificar un ave pues estará mejor entrenado.

**Resultados**

Dado que el modelo requiere de muchas mejoras, los resultados mostrados no concluyen ser los más acertados. Las imágenes procesadas por la Red Neuronal Convolucional (CNN) permitieron que el modelo aprendiera características distintivas de cada especie, pero no lo suficiente como para poder clasificar cada ave con tanta precisión.

Las predicciones del modelo se evaluaron utilizando un conjunto de imágenes de prueba.  
  
**Conclusiones**  
  
En resumen, el desarrollo del sistema de detección para la clasificación de especies de aves implicó varios pasos cruciales. Inicialmente, se recopilaron y limpiaron cuidadosamente las imágenes de aves para garantizar la calidad y relevancia de los datos. La creación del modelo involucró el diseño de una Red Neuronal Convolucional (CNN) con capas convolucionales, de normalización, de agrupamiento y densas, adecuadamente configuradas para la tarea de clasificación. La implementación de técnicas de aumento de datos y la normalización mediante **ImageDataGenerator** fueron fundamentales para mejorar la generalización del modelo. El entrenamiento del modelo se optimizó mediante el uso del algoritmo Adam y la incorporación de un callback de detención temprana para evitar el sobreajuste.

Los resultados mostraron ser no tan acertados. Pero para fin del proyecto y como entendimiento de cómo es que funcionan las redes neuronales convolucionales, los resultados fueron muy valiosos. Esto es porque con el desarrollo de la arquitectura del modelo y del proyecto como tal, pude ver más a detalle cómo es que funcionan estos y la importancia de sus capas. Es un proceso bastante complejo el que se lleva a cabo para poder identificar factores dentro de una imagen que, para los humanos es bastante simple, pero que para las computadoras se les tiene que entrenar.

Como recomendaciones a futuras mejoras, sería tener un modelo mejormente entrenado, tener un dispositivo que pueda cargar y entrenar un modelo de manera masiva o con gran cantidad de imágenes. Contar con más tiempo para poder llevar a cabo muchas más pruebas de entrenamiento y así poderlas modificar para resultar con un mejor modelo.