

INSTITUTO SUPERIOR POLITÉCNICO CÓRDOBA

MÓDULO: CIENTÍFICO DE DATOS

PROFESORES:

- Facundo OLIVA CÚNEO
- Carlos Ignacio CHARLETTI
- Moisés TINTE



*“Muéstrame una imagen
y
te diré tu destino”*

**Desarrollo de un Modelo de Red
Neuronal Convolutacional (CNN) para
Clasificación de Imágenes**

GRUPO 18

Mariano LEDEZMA

Natalia LAMIA

Viviana FARABOLLINI

Índice

Introducción	3
Desarrollo.....	4
I – Pasos en el Proceso de Desarrollo del Modelo CNN.....	4
Configuración del Entorno	
Conjunto de Datos	
Preprocesamiento de Datos	
Definición del Modelo	
Entrenamiento del Modelo	
Evaluación del Modelo	
Recomendaciones Futuras	
II – Análisis Detallado de Componentes Técnicos	6
1. Aprendizaje Automático	
2. Desarrollo de Sistemas de IA	
3. Técnicas de Procesamiento del Habla	
4. Procesamiento de Imágenes	
5. Minería de Datos	
III – Información Técnica Adicional.....	9
1. Arquitectura del Modelo CNN	
2. Entrenamiento y Optimización	
3. Evaluación del modelo	
4. Generación de Texto a Voz	
5. Funcionalidad de la Aplicación	
IV – Aplicación del Modelo con Data Set Alternativo y Mejora de Funcionalidad	
Desarrollando Interfaz de Usuario.....	10
1. Construcción de una Base de Datos utilizando Web Scraping para aplicar el Modelo CNN	
2. Desarrollo de Interfaz de Usuario utilizando Procesamiento de Voz	
Conclusión.....	13
Bibliografía.....	14
Anexos.....	14

Introducción

El presente informe documenta el proceso de desarrollo de un modelo de Red Neuronal Convolutacional (CNN) para la clasificación de imágenes.

Se ha perseguido el objetivo de crear un modelo capaz de clasificar imágenes con alta precisión con la intención de avanzar en la construcción de una aplicación que puedan utilizar los usuarios de una agencia de turismo para recibir sugerencias de destinos turísticos en Argentina de manera ágil y dinámica.

Para ello se configuró un entorno de trabajo que permita desarrollar, ejecutar el código y almacenar archivos e interactuar con el equipo de trabajo. En esta ocasión se utilizó Kaggle, Google Colab y Google Drive. Además se aprovechó la aceleración proporcionada por la Unidad de Procesamiento Tensorial (TPU).

Se utilizó Python y librerías numpy, os, tensorflow, keras, seaborn, matplotlib, gTTS.

Se trabajó con una base de datos aportada por Kaggle y se intentó construir una base de datos utilizando scraping adaptando el modelo a ambas.

Finalmente, se incluyó la posibilidad de cargar imágenes para predecir y se diseñó un script para dar sugerencias de destinos turísticos por voz, se desarrolló una GUI como experiencia de prueba para la interacción con usuarios que se acompaña con una base de imágenes en MySQL.

Desarrollo

Detallaremos los pasos realizados en el proceso de desarrollo de un modelo de Red Neuronal Convolutacional (CNN) para la clasificación de imágenes.

Trabajamos en un contexto de aprendizaje supervisado. El aprendizaje supervisado es un tipo de aprendizaje automático en el cual se utiliza un conjunto de datos etiquetados para entrenar el modelo. Cada entrada del conjunto de datos tiene una etiqueta asociada, que actúa como una "respuesta" que el modelo debe aprender a predecir. Según Müller y Guido, *"en el aprendizaje supervisado, se le muestra al modelo ejemplos etiquetados durante el entrenamiento, para que pueda aprender a predecir el resultado correspondiente para datos nuevos y no etiquetados"* (Müller & Guido, 2016).

4

Durante el entrenamiento, se fueron ajustando sus parámetros para minimizar la discrepancia entre las predicciones y las etiquetas reales de los datos de entrenamiento.

En este caso, el modelo CNN se entrenó utilizando un conjunto de datos de imágenes de escenas naturales ('buildings', 'forests', 'glaciers', 'mountains', 'sea' y 'streets') proporcionado por Kaggle, donde cada imagen está etiquetada con una de las seis categorías mencionadas. El objetivo del entrenamiento ha sido que el modelo aprenda a asociar características visuales específicas de las imágenes con las etiquetas correspondientes.

En una segunda parte presentaremos un análisis de componentes técnicos relacionados a cada uno de los espacios del Módulo Científico de Datos de la Tecnicatura en Ciencia de Datos e Inteligencia Artificial del ISPC.

Sumaremos, también, alguna información técnica adicional.

Por último, expondremos dos intentos relacionados con el modelo: uno, de aplicar el modelo con una base de datos construida con web scraping, respetando la estructura de la de kaggle pero con imágenes de argentina, modeladas y etiquetas. Intento que ha mostrado limitaciones; y, dos, desarrollar una GUI que para mejorar la interacción con los usuarios en la que hay que trabajar el contenido de la respuesta por voz.

En el anexo se adjuntan los archivos correspondientes con las distintas versiones desarrolladas del proyecto: Versión I -Construcción del modelo-, Versión II -Construcción de Base de Datos con web scraping, prueba con el modelo CNN, Versión III – Proyecto final, Modelo CNN con predicción de imágenes de usuario y respuesta de sugerencias de destino turístico por voz, GUI -desarrollo de una aplicación de prueba con usuarios-.

I - Pasos en el Proceso de Desarrollo del Modelo CNN

Configuración del Entorno: Se utilizó la plataforma Kaggle, Google Colab y Google Drive para desarrollar, ejecutar el código y almacenar archivos. Se configuró el entorno para aprovechar la aceleración proporcionada por las Unidades de Procesamiento Tensorial (TPU), lo que permite un entrenamiento más rápido del modelo.

Conjunto de Datos: Trabajamos con un conjunto de datos proporcionado por Kaggle: ¿Intel Image Classification Dataset¹. Los datos contienen imágenes de escenas naturales de todo el mundo, con alrededor de 25,000 imágenes de tamaño 150x150 distribuidas en seis categorías: 'buildings', 'forest', 'glacier', 'mountain', 'sea' y 'street'. Los datos de entrenamiento, prueba y predicción están separados en archivos zip con alrededor de 14,000 imágenes en Train, 3,000 en Test y 7,000 en Prediction.

Preprocesamiento de Datos: Se utilizaron las bibliotecas TensorFlow y Keras para cargar y preprocesar los datos de imágenes. Se dividieron los datos en conjuntos de entrenamiento, validación y prueba. Además, se aplicaron técnicas de aumento de datos para acrecentar la cantidad de muestras de entrenamiento y mejorar la generalización del modelo.

La elección de Keras y Tensorflow para desarrollar este trabajo fue porque según Aurélien Géron *"Keras es una API de alto nivel para redes neuronales, escrita en Python y capaz de ejecutarse sobre TensorFlow, CNTK, o Theano. Su diseño modular y simplicidad la hacen ideal para la rápida experimentación y prototipado, mientras que TensorFlow provee una infraestructura robusta para la implementación y escalabilidad de modelos de machine learning"* (Géron, 2022).

Definición del Modelo CNN: Se definió una arquitectura de red neuronal convolucional (CNN) utilizando la API secuencial de Keras. La arquitectura consta de varias capas convolucionales y de agrupación, seguidas de capas completamente conectadas. Se utilizó la función de activación ReLU en las capas convolucionales y se aplicó regularización con dropout para evitar el sobreajuste.

Entrenamiento del Modelo: El modelo se entrenó utilizando el conjunto de datos de entrenamiento y se validó utilizando el conjunto de datos de validación. Se utilizó el optimizador Adam y la función de pérdida de entropía cruzada categórica para optimizar el modelo. El entrenamiento se realizó durante 30 epochs, con un historial de entrenamiento para monitorear la pérdida y la precisión en cada una.

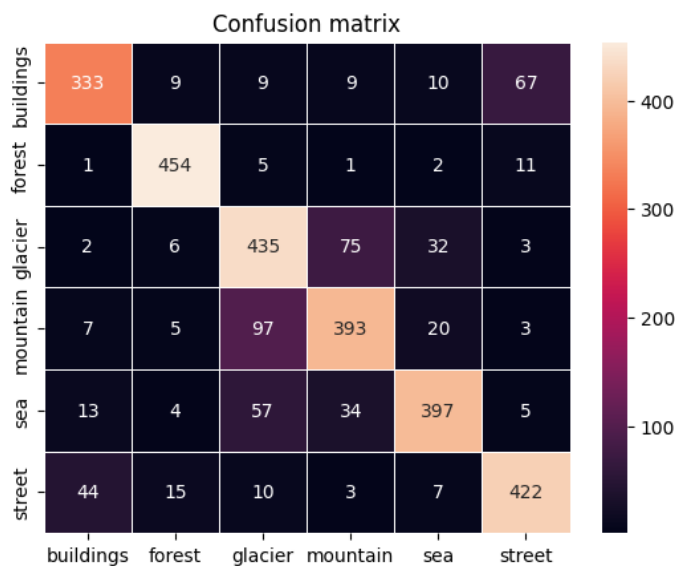
Evaluación del Modelo: El modelo se evaluó utilizando el conjunto de datos de prueba para calcular la pérdida y la precisión en datos no vistos durante el entrenamiento. Además, se realizaron pruebas adicionales visualizando las predicciones del modelo en imágenes de prueba seleccionadas.

La respuesta a `model.evaluate(validation_generator)` fue la siguiente:

```
94/94 [=====] - 3s 29ms/step - loss: 1.1895 - accuracy: 0.8113
```

¹ "Kaggle Intel Image Classification": link: <https://www.kaggle.com/puneet6060/intel-image-classification/data>

Este resultado indica que el modelo alcanzó una pérdida de 1.1895 y una precisión del 81.13% en el conjunto de datos de prueba. El modelo entrenado demostró un alto rendimiento en la clasificación de imágenes, con una precisión superior al 80% en el conjunto de datos de prueba. Esto sugiere que el modelo es capaz de generalizar bien a datos nuevos y no vistos durante el entrenamiento. Además, las pruebas visuales mostraron que el modelo puede realizar predicciones precisas en imágenes de prueba seleccionadas. Sin embargo, confunde en algunas ocasiones montañas con edificios, calles con edificios, mar con glaciar. Esta información, junto con la visualización de la matriz de confusión, proporciona una comprensión más completa del rendimiento del modelo y áreas potenciales para mejoras futuras.



Recomendaciones Futuras: Se sugiere explorar técnicas adicionales de mejora del rendimiento del modelo, como la optimización de hiperparámetros, la exploración de arquitecturas de red más complejas y la aplicación de técnicas avanzadas de aumento de datos. Además, se recomienda realizar pruebas exhaustivas en un entorno de producción antes de implementar el modelo en una aplicación real.

II - Análisis Detallado de Componentes Técnicos

Abordaje de componentes técnicos desde cada asignatura.

1. Aprendizaje Automático

El aprendizaje automático (Machine Learning) se centra en el desarrollo de algoritmos que permiten a las computadoras aprender de y hacer predicciones o decisiones basadas en datos.

Aplicación en el Proyecto:

Preprocesamiento de Datos

Normalización de imágenes: Rescalado de los píxeles de las imágenes para que sus valores estén en el rango [0, 1].

Aumento de datos: Técnicas como rotaciones, traslaciones, y flips para aumentar la variedad del conjunto de datos de entrenamiento y mejorar la capacidad de generalización del modelo.

Construcción del Modelo

Definición de una arquitectura CNN con capas convolucionales, de pooling, y completamente conectadas.

Uso de funciones de activación como ReLU y técnicas de regularización como Dropout.

Entrenamiento del Modelo:

Optimización con Adam y uso de la función de pérdida de entropía cruzada categórica. Monitoreo del rendimiento del modelo mediante el cálculo de métricas de precisión y pérdida en el conjunto de validación.

2. Desarrollo de Sistemas de IA

El desarrollo de sistemas de IA implica la creación de soluciones que pueden realizar tareas que normalmente requieren inteligencia humana, como el reconocimiento de patrones, la toma de decisiones y la predicción.

Aplicación en el Proyecto:

Integración del Modelo

Uso de la librería TensorFlow/Keras para construir, entrenar y evaluar la CNN.

Guardado del modelo y sus pesos en un archivo con extensión .h5 utilizando la función `model.save_weights('/kaggle/working/model_weights_cnn.h5')` de Keras, lo cual permite preservar el estado del modelo para futuras cargas y uso sin necesidad de volver a entrenarlo

Despliegue y Uso en una Aplicación:

Uso de APIs para integrar el modelo entrenado en una aplicación interactiva que sugiere destinos turísticos basándose en imágenes proporcionadas por los usuarios.

3. Técnicas de Procesamiento del Habla

El procesamiento del habla se refiere a la tecnología que permite a las máquinas comprender y generar lenguaje hablado.

Aplicación en el Proyecto:

Generación de Texto a Voz (TTS):

Uso de la librería gTTS (Google Text-to-Speech) para convertir las predicciones del modelo en mensajes de audio que informan al usuario sobre la categoría de la imagen y proporcionan sugerencias turísticas relacionadas.

4. Procesamiento de Imágenes

El procesamiento de imágenes involucra técnicas para la manipulación y análisis de imágenes digitales con el fin de mejorar su calidad, extraer información relevante significativa y preparar las imágenes para su uso en aplicaciones avanzadas.

Aplicación en el Proyecto:

Cargar y Preprocesar Imágenes:

Redimensionamiento de imágenes a un tamaño uniforme (150x150) y su etiquetado para su uso en el modelo CNN.
Conversión de las imágenes a arrays de numpy y normalización.

Visualización de Resultados:

Uso de matplotlib para mostrar imágenes de entrenamiento y sus etiquetas de clase correspondientes, facilitando la inspección de los datos de entrenamiento y la verificación de carga correcta.
De manera similar se procesan las imágenes de prueba, incluyendo carga de imágenes, normalización de los valores de píxeles y la redimensión de las imágenes para que tenga el mismo tamaño que las imágenes de entrenamiento.

5. Minería de Datos

La minería de datos es el proceso de descubrir patrones, tendencias y relaciones en grandes conjuntos de datos utilizando métodos automáticos y algorítmicos. Esto permite extraer información valiosa que puede ser utilizada para tomar decisiones informadas.

Aplicación en el Proyecto:

En el contexto de nuestro proyecto, la minería de datos puede ser aplicada en varios aspectos clave para mejorar la funcionalidad y el rendimiento del sistema de

procesamiento de imágenes y voz. A continuación, se describen algunos de estos aspectos:

Análisis de Datos de Imágenes:

Transformación y Preprocesamiento: Antes de entrenar el modelo CNN, las imágenes pasan por un proceso de preprocesamiento, donde se aplican transformaciones como el cambio de tamaño, normalización y aumento de datos. Estas técnicas ayudan a mejorar la calidad del modelo y a reducir el overfitting.

Extracción de Características: Técnicas de extracción de características, como histogramas de imágenes, detección de bordes y segmentación, se utilizan para identificar y analizar patrones en los datos de imágenes. Estos patrones son esenciales para el entrenamiento efectivo del modelo CNN.

Evaluación y mejora del Modelo:

Visualización de la Matriz de Confusión: Una vez entrenado el modelo, se utiliza la matriz de confusión para evaluar su rendimiento. La matriz de confusión ayuda a identificar qué clases de imágenes el modelo clasifica correctamente y cuáles no, proporcionando información valiosa para ajustar y mejorar el modelo.

Análisis de Errores: Al analizar los errores del modelo, se pueden descubrir patrones en las imágenes que el modelo tiene dificultades para clasificar. Este análisis permite realizar ajustes específicos en el preprocesamiento de datos o en la arquitectura del modelo.

III - Información Técnica Adicional

1. Arquitectura del Modelo CNN:

Capas Convolucionales: Extraen características locales de las imágenes mediante filtros aplicados con convoluciones.

Capas de Pooling (MaxPooling2D): Reducen la dimensionalidad de las características extraídas, reteniendo la información más importante y ayudando a controlar el sobreajuste.

Capas Completamente Conectadas (Dense): Actúan como un clasificador en la parte final del modelo, procesando las características extraídas para realizar la clasificación final.

Función de Activación ReLU: Introduce no linealidad al modelo, permitiendo aprender representaciones más complejas.

Dropout: Técnica de regularización que ayuda a prevenir el sobreajuste al apagar aleatoriamente neuronas durante el entrenamiento.

2. Entrenamiento y Optimización:

Optimizador Adam: Combina las ventajas de dos métodos de optimización: AdaGrad y RMSProp. Es eficiente y requiere menos memoria.

Pérdida de Entropía Cruzada Categórica: Es adecuada para problemas de clasificación multiclase, calculando la discrepancia entre las distribuciones de probabilidad predicha y real.

10

3. Evaluación del Modelo:

Precisión: Proporción de predicciones correctas realizadas por el modelo.

Matriz de Confusión: Permite visualizar el rendimiento del modelo en cada clase, mostrando aciertos y errores en una matriz cuadrada.

4. Generación de Texto a Voz:

Librería gTTS: Herramienta que permite convertir texto a audio utilizando los servicios de Google Text-to-Speech, compatible con múltiples idiomas.

5. Funcionalidad de la Aplicación

Tkinter: (framework) o biblioteca gráfica para Python que se utiliza para desarrollar interfaces gráficas de usuario (GUI, por sus siglas en inglés).

IV – Aplicación del Modelo con Data Set alternativo y mejora de funcionalidad desarrollando interfaz de usuario

1. Construcción de una Base de Datos utilizando web scraping para aplicar el modelo CNN.

En el contexto del proyecto se intenta generar una base de datos propia para ser adaptada al modelo CNN.

Su principal objetivo fue obtener y etiquetar imágenes de diversas regiones de Argentina desde páginas web, específicamente desde Shutterstock.

Se utilizaron técnicas de web scraping para extraer imágenes de la página web antes mencionada. Posteriormente se etiquetaron con el nombre de la región correspondiente y se almacenaron en Google Drive. El proceso se llevó a cabo en Google Colab para aprovechar su entorno colaborativo y la integración con Google Drive.

Se creó una estructura de carpetas para almacenar las imágenes organizadas por región: Patagonia, Noroeste, Noreste, Cuyo, Buenos Aires y Córdoba. Se iteró sobre las diferentes regiones, descargando y etiquetando las imágenes con el nombre correspondiente de su carpeta contenedora.

El proceso de scraping y etiquetado de imágenes se realizó exitosamente, proporcionando un dataset adecuado pero no suficiente en cantidad de imágenes como para entrenar el modelo. Este dataset podría llegar a ser fundamental para el desarrollo de la futura aplicación de turismo en tanto y en cuanto se mejore en cantidad de imágenes almacenadas.

Por ello, se decide continuar con el dataset proporcionado por Kaggle dejando abierta la posibilidad de construir una base de datos adecuada.

2. Desarrollo de Interfaz de Usuario utilizando Procesamiento de Voz

Con la intención de mostrar la predicción del modelo a carga de imágenes simulando lo que podría ser, con modificaciones, la interacción con el usuario a través de respuesta de voz, se creó una GUI (Interfaz Gráfica de Usuario).

Para ello se ha utilizado Tkinter, una biblioteca de Python que permite diseñar y crear interfaces gráficas para programas con la que se puede construir ventanas, botones, cuadros de texto, menús y otros elementos visuales que hacen que el modelo sea interactivo y fácil de usar a través de un visor de imágenes.

Esta aplicación también interactúa con una base de datos MySQL y reproduce audio con pygame.

Su funcionalidad principal es la de *cargar imagen* desde el sistema de archivos del usuario, mostrándola a la izquierda del panel; *predecir imagen* utilizando un módulo externo para predecir la categoría de la imagen cargada y actualiza la etiqueta correspondiente; *recomendar imágenes* basándose en la categoría predicha, obtiene imágenes de la base de datos y las muestra en un conjunto de subplots en el lateral derecho del panel; y, *reproducir audio* de un archivo asociado a la predicción.

La ventana principal de la GUI admite cargar imágenes, predecir su categoría y ver imágenes recomendadas relacionadas. La conexión a la base de datos permite obtener datos dinámicos basados en la predicción del modelo CNN.

Además, y con el mismo propósito el equipo se propone desarrollar un script que utilice reconocimiento de voz para procesar las solicitudes de viaje. Esta funcionalidad permitiría a los usuarios realizar solicitudes de viaje de manera intuitiva simplemente utilizando su voz, eliminando la necesidad de escribir comandos o hacer clic en opciones específicas. Esto facilitaría el uso de la aplicación, especialmente para

aquellos usuarios que pueden tener dificultades con la entrada de texto tradicional, como personas con capacidades diferentes o aquellos que prefieren una experiencia más fluida y práctica.

Conclusión

El desarrollo de un modelo de Red Neuronal Convolutacional (CNN) para la clasificación de imágenes ha permitido avanzar en la construcción de una aplicación de turismo interactiva y eficiente.

El uso de Técnicas de Aprendizaje Automático, Desarrollo de Sistemas de IA, Procesamiento de Imágenes y Voz, y Minería de Datos ha sido crucial para alcanzar los objetivos del proyecto.

Aunque el modelo alcanzó una precisión del 81.13%, se identificaron áreas de mejora, como la optimización de hiperparámetros y la exploración de arquitecturas de red más complejas. Además, se ha probado que la integración de reconocimiento de voz podría mejorar significativamente la interacción del usuario con el desarrollo de una aplicación.

Los futuros trabajos deberán centrarse en la ampliación del dataset, optimización del modelo y pruebas exhaustivas en un entorno de producción para asegurar su robustez y escalabilidad en una aplicación real. Este proyecto sienta las bases para una herramienta innovadora en el ámbito del turismo, ofreciendo recomendaciones personalizadas de destinos turísticos en Argentina mediante una experiencia de usuario mejorada y eficiente.

Por el momento y hasta aquí, podemos decir con certeza que los avances nos muestran que es plenamente viable continuar con el proyecto. Los resultados obtenidos no solo demuestran el potencial de esta aplicación, sino que también abren la puerta a numerosas oportunidades para expandir sus capacidades. La implementación de mejoras sugeridas y la incorporación de nuevas tecnologías pueden llevar esta herramienta a un nivel superior, beneficiando a usuarios y potencialmente transformando la manera en que se interactúa con información turística. La viabilidad técnica y el impacto positivo observados respaldan firmemente la continuidad y expansión del proyecto.

‘Muéstrame una imagen y te diré tu destino’

Bibliografía

Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: A guide for data scientists*. Sebastopol, CA: O'Reilly Media, Inc.

Géron, A. (2022). *Aprende Machine Learning con Scikit-Learn, Keras y TensorFlow: Conceptos, herramientas y técnicas para construir sistemas inteligentes*. Sebastopol, CA: Anaya Multimedia.

Anexo

En esta sección se incluyen los enlaces para consultar los scripts utilizados en la construcción del modelo CNN y su programa en distintas versiones. Estos scripts están alojados en Google Drive y GitHub:

- Google Drive:
https://drive.google.com/drive/folders/1AbHcqU13pLmIQ0IVJlkj8SXDfcjdsDfS?usp=drive_link
 - ✓ Versión I:
 - Primer Evidencia de Aprendizaje: Pre-Proyecto
 - I - ClasificacionFotos_ConRtaVoz_Con_Informe_Técnico.ipynb : Desarrollo del Modelo CNN (Clasificación de Imágenes utilizando Redes Neuronales y explicación paso a paso)
 - Informe Técnico: Desarrollo de un Modelo de Red Neuronal Convolutiva (CNN) para Clasificación de Imágenes.
 - ✓ Versión II:
 - CRISP-DM AppTurismo
 - II – ScrapingTurismo.ipynb : Probando construir un dataset de imágenes de zonas turísticas de Argentina realizando scraping a <https://www.shutterstock.com/>
 - Turismo_Argentina: 'dataset_path' proveniente de ScrapingTurismo
 - III – ClasificaciónFotosTurismoArg_ConRtaVoz.ipynb: aplicación del modelo en otra base de datos, en este caso 'Turismo_Argentina'
 - ✓ Versión III:
 - TPF_ModeloCNN_CargaImágenes y RtaVoz.ipynb
 - ✓ GUI:
 - detalle-bd_recomendaciones.png
 - generar_bd_mysql
 - imágenes.py
 - Carpetas:
 - images
 - img_subida
 - seg_pred
 - seg_test
 - seg_train

- GUI.py
 - Model_weights_cnn.h5
 - ✓ TPF_Informe Final Módulo Científico de Datos Grupo 18
 - ✓ Presentación Grupo 18
 - ✓ GUI_funcionando
- GitHub:
<https://github.com/natalialamia/GRUPO-18--ISPC--CIENTIFICO-DE-DATOS--2024>