MUSHROOMS CLASIFICATION

Uri Jared Gopar Morales, ITC A01709413, Tecnologico Campus Querétaro.

***Abstracto*- En el siguiente documento se presentará el análisis de la clasificación de diferentes tipos de hongos.**

I. INTRODUCCION

En nuestro planeta existen números organismos que cumplen con diferentes funciones para el ecosistema. Uno de estos organismos son los hongos, los cuales cumplen su función como descomponedores hasta simbiontes. Sin embargo, la identificación de todas las especies de hongos conocidas es compleja, debido a que varias de estas especies presentan características muy similares, lo que hace que la clasificación de forma manual sea tardada y propensa a errores. En este documento, se utilizará un dataset de imágenes de hongos, el cual contiene varias especies clasificadas por su especie.

El objetivo principal es entrenar un modelo de redes neuronales convolucionales (CNN) para clasificar de manera automática las especies de hongos a partir de las imágenes.

II. MANEJO DE DATOS

El dataset cuenta con un total de 7,763 imágenes de hongos, las cuales están divididas por 11 especies. Para poder cargar dichas imágenes se utilizó Pytorch el cual nos proporciona una herramienta que mapea automáticamente las subcarpetas a etiquetas de clases, esto quiere decir que si las imágenes se encuentran en una carpeta que contiene el nombre de (AMANITA), el label de todas esas imágenes será AMANITA. Adicionalmente inspeccionando el dataset me tope que existían imágenes dañadas lo que quiere decir que no se veía nada o estaban en otro formato, para identificar de una manera más eficiente dichas imágenes cree una clase personalizada la cual me decía cuales eran sospechosas.

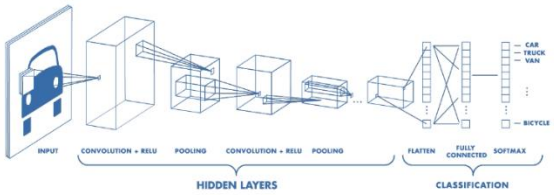
Una vez con todas las imágenes sin errores, fueron procesadas mediante una serie de transformaciones definidas en pipeline,

- Se redimensionaron a un tamaño de 128x128 pixeles, esto principalmente para que todas sean de la misma dimensión.
- Se convirtieron a Tensores, el cual es el formato requerido por Pytorch para alimentar los datos al modelo.

- Se normalizaron en un rango de valores entre (-1,1) utilizando los valores medio y desviaciones estándar de los canales de color RGB, para mejorar la convergencia.

III. MODELO

Dicho anteriormente el modelo que decide utilizar fue un modelo de redes neuronales convolucionales (CNN), ya que la arquitectura de este modelo es específicamente para procesar datos como imágenes. Este modelo tiene la capacidad para poder captar patrones espaciales y jerárquicos de las imágenes (bordes, texturas y diferentes formas) a través de sus capas convolucionales.



IV. IMPLEMENTACION

Capas convolucionales: Las capas aplican los filtros “Kernels” sobre la imagen de entrada para detectar características, bordes, texturas y detalles. Utilice 2 capas convolucionales:

- La primera convierte la imagen de entrada en un mapa de características con 32 filtros de 3x3.
- La segunda capa toma la salida de la primera y genera 64 filtros adicionales.

Se utiliza la función de activación ReLU, para poder introducir no linealidades en el modelo, esto para aprender relaciones mas complejas en los datos.

Capas de Max Pooling reducen el tamaño de las representaciones generadas por las capas convolucionales al seleccionar el valor máximo en una región. Ayuda a reducir la dimensionalidad, haciendo el modelo mas eficiente, y a la vez ayuda para resaltar las características mas importantes detectadas por las capas convulsiónales.

Capas conectadas, estas juntan todas las características extraídas por las capas convolucionales con las salidas del modelo. En mi primera capa conectada de mi modelo contiene 512 neuronas. Mi segunda capa lo que hace es que conecta al numero de clases de salida, los tipos de hongo a los que corresponde.

Por terminar tenemos la función de perdida, la cual mide la diferencia entre predicciones del modelo y ajusta los parámetros para mejorar sus predicciones.

V. ENTRENAMIENTOS

Para implementar el proceso de entrenamiento de mi red neuronal convolucional, separe mi dataset en 60% entrenamiento, 20% valores de validación y 20% prueba, con la finalidad para poder ver como es el rendimiento de mi modelo y como este a través de las épocas va aprendiendo.

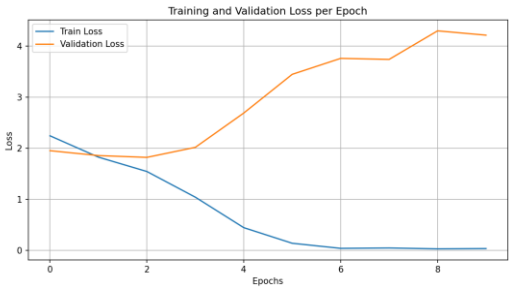
- Entrenamiento 60%: nos ayuda para ajustar sus pesos y parámetros, minimizando la perdida por época.
- Validación 20%: Evalúa al modelo durante su entrenamiento, este ajusta

los hiperparámetros como la tasa de aprendizaje con datos que nunca ha visto.

- Prueba 20%: Se utiliza para evaluar el rendimiento final del modelo, este nos ayuda para ver qué tan acertado es nuestro modelo con las predicciones de datos que nunca ha visto.

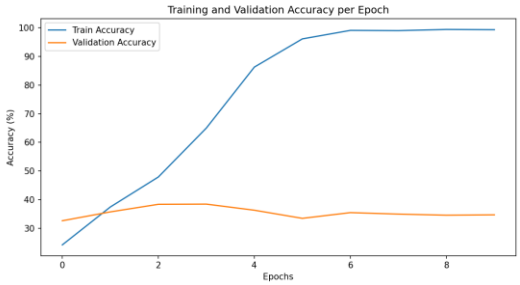
VI. RESULTADOS

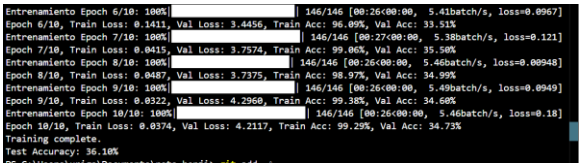
Los resultados obtenidos a través de dicho modelo fueron del 0.3610 lo que nos indica que el modelo clasifica de modelo correcta el 36.10% de las especies de hongos.



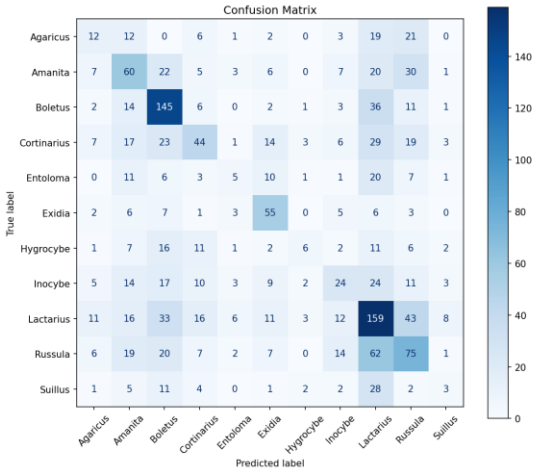
Sin embargo, al ver la grafica anterior, la cual se muestra el training y validetion Loss entre las épocas. Observamos una disminución en términos de la perdida en entrenamiento lo cual nos indica que el modelo se está ajustando. Pero al momento de visualizar la gráfica de Validetion Loss no sigue el mismo patrón de perdida, en cambio se ve como este al principio está disminuyendo y después de la segunda época aumenta drásticamente. Esto es un indicio que el modelo cuenta con overfitting, al ajustarse demasiado a los datos de entrenamiento pierde la capacidad para generalizar bien a datos no vistos como son los de validetion.

Podemos ver la precisión del modelo en entrenamiento y validetion, de igual manera, se observa como en los datos de entrenamiento el modelo se ajusta rápidamente, mientras para los datos de validation se mantiene bajo. Por lo que muestra una validation acc de 34.72% y un test accuracy de 36.10%.





Para observar mejor como el modelo está clasificando, realicé una matriz de confusión, de este modo me pude dar cuenta de cuales clases tienen mejor rendimiento y en cuales se confunde el modelo. Como resumen rápido podemos ver que el modelo tiene un buen rendimeito para la clasificación de (Boletus y Lactarius). Y las clases con mayor sesgos son Agarcus y Amanita, dejando que estas clases son las que cuentan con características mas parecidas entre si por lo que es complicado predecir.



VII. MEJORA AL MODELO

Regularización: Implementar técnicas como la regularización L2 o el dropout podría ayudar a reducir el sobreajuste, haciendo que el modelo dependa menos de los patrones específicos de los datos de entrenamiento.