

## 1. Pasos necesarios a seguir.

1.1 Los modelos aprenden a través de ejemplos, es decir, el modelo aprenderá con imágenes similares. Es decir, se deben identificar claramente qué es lo que queremos categorizar. En este punto es importante elegir un conjunto de imágenes que tengan características similares que serán las que utilizará el modelo para su entrenamiento.

1.2 Una vez que contamos con dicha información, se procede al etiquetado manual de las imágenes, existen algunas herramientas como las brindadas por Azure o Ultralytics que nos permiten etiquetar mediante recuadros de selección.

1.3 Luego, con el conjunto de datos etiquetados, se procede a la creación de conjuntos de datos de train, validation y test (como en cualquier modelo de ML).

1.4 Podemos obtener los pesos de librerías open source o de servicios de pago, y comenzamos a entrenar el modelo a partir de dicho modelo preentrenado.

1.5 Finalmente, se realiza un seguimiento de las métricas obtenidas por el modelo y se realizarán ajustes en caso de ser necesario.

## 2. Descripción de posibles problemas que puedan surgir y medidas para reducir el riesgo.

Algunos problemas que nos podemos encontrar al momento de entrenar nuestro modelo:

2.1 No contar con suficientes datos de las categorías sobre las cuales queremos entrenar nuestro modelo.

2.2 Overfitting, el modelo puede aprender de los datos de entrenamiento y no generalizar para nuevas categorías.

2.3 Problemas al momento de realizar el etiquetado, dada la naturaleza de la información que puedan confundir al modelo.

2.4 Imbalance de clases, que puede llevar al modelo a tomar a sobreajustarse a la clase mayoritaria generando sesgos y pérdida de información.

## 3. Estimación de cantidad de datos necesarios así como de los resultados, métricas, esperadas.

La cantidad de datos necesarios pueden variar dependiendo de factores como la complejidad del problema, la arquitectura seleccionada para resolverlo, la cantidad de ruido que tenga nuestro set de datos y la diversidad de la información recolectada. Sin embargo, es buena práctica contar con una cantidad del orden de mil datos etiquetados, para ser considerado un buen punto de partida.

Para este tipo de problemas los modelos normalmente se benefician de tener mas información, ya que pueden prevenir el overfitting y mejorar la generalización para nuevos datos. Si el set de datos es pequeño, el modelo puede tender a memorizar la información en lugar de encontrar patrones empeorando el rendimiento en nuevos datos.

La metrica de desempeño mas importante para un modelo de clasificación es el 'accuracy' sobre nuestros valores de validación. Existen otras métricas que habrá que analizar en detalle dependiendo el tipo de problema. El accuracy mide el porcentaje de imagenes clasificadas correctamente (tener cuidado si nuestras clases no están balanceadas). Buenos valores de esta metrica estan considerados entre 70 - 90 % que se corresponden con valores acordes a una problemática real. Es buena practica comparar el desempeño de nuestro modelo con modelos base y modelos en estado del arte, para determinar que tan bien lo está haciendo nuestro modelo y detectar posibles mejoras.

4. Enumeración y pequeña descripción (2-3 frases) de técnicas que se pueden utilizar para mejorar el desempeño, las métricas del modelo en tiempo de entrenamiento y las métricas del modelo en tiempo de inferencia.

- Transfer Learning: utilizar modelos preentrenados como punto de partida, como ResNet50, COCO, EfficientNet, etc. para aprovechar el conocimiento y las cualidades aprendidas por un modelo mas grande y adaptarlo a un caso en particular.
- Data Augmentation: forma de optimizar el reconocimiento de objetos realizando transformaciones, como recortar, voltear, rotar, escalar o agregar ruido. Permite aumentar la cantidad y diversidad de datos de entrenamiento.
- Hyperparameter tuning: experimentar con diferentes valores de hiperparámetros (loss, precision, accuracy o recall) o utilizar diferentes técnicas como grid search, random search, etc. para encontrar la mejor combinación de hiperparámetros.
- Cuatización: por este método se intenta pasar las representaciones en un punto flotante (de pesos, y a veces activaciones) a formatos de menor precisión, pasando numeros con menor cantidad de bits, facilitando así, los cálculos.
- Pruning o poda: se basa en la eliminación de conexiones o neuronas poco significativas en las predicciones. Se analizan las magnitudes, aquellas mas cercanas a cero son considerados menos importante, por tanto, eliminados.
- Destilación de conocimiento: aplicable a redes de clasificación, consiste en generar un red 'alumno' que aprenda a replicar el comportamiento de una red mas compleja 'profesor'.