Actividad Integradora

Cada equipo diseñará un clasificador de imágenes, las categorías que clasificará cada equipo son las siguientes:

Equipo #1 y #4: Clasificador de 5 vehículos de transporte (i.e. tren, avión, carro, camión, bicicleta)

Equipo #2 y #5: Clasificador de 5 alimentos diferentes (i.e. ensalada, atún, pasta, hamburguesa, pizza)

Equipo #3: Clasificador de 5 animales diferentes (i.e. mono, león, hormiga, tortuga, búho)

En su diseño y construcción del modelo se evaluarán equitativamente los siguientes 4 aspectos:

1. **Recolección y procesamiento de datos**

   Detallen cuáles fueron las técnicas de recolección empleadas para construir su set de entrenamiento, prueba y validación. Grafiquen los datos recolectados y determinen si su distribución y cantidad son apropiadas para entrenar su modelo.

   Nota: En caso de que la distribución de los datos recolectados no sea equitativa, mencionen las técnicas de aumentación de datos que usaron. El uso de técnicas de aumentación de datos será tomado en cuenta para puntos extra.

2. **Selección y entrenamiento de modelos**

   El modelo base que estarán utilizando para la clasificación de imágenes será VGG16, a la cual le harán *fine-tuning.* Detallen su plan de entrenamiento indicando los hiper parámetros, la función de costo y el método para ajustar parámetros. Mencionen las técnicas para evitar el sobre y sub ajuste que incorporaron en su modelo o técnica de entrenamiento.

   Nota: El entrenar un modelo adicional a VGG16, será tomado en cuenta para puntos extra.

3. **Evaluación y métricas de desempeño**

   Identifiquen cuáles son las métricas más apropiadas para medir el desempeño de su modelo. Grafiquen el desempeño de sus modelos respecto a las épocas de

entrenamiento, incluyan la gráfica de entrenamiento y validación. Construyan matrices de confusión para evaluar los clasificadores que diseñaron.

4. **Despliegue y reproducibilidad del modelo**

   Construyan un repositorio de github como control de versiones del proyecto. El repositorio debe incluir un requirement.txt y un environment.yml para replicar el ambiente que fue utilizado para ejecutar el código. El .ipynb será revisado en Google colab por lo que asegúrense de que corra en ese ambiente. Incluyan un README que contenga la documentación apropiada para reproducir sus experimentos. Sigan las indicaciones de https://help.github.com/en/github/creating-cloning-and-archiving-repositories/about-readmes para generar un README file apropiado.

Para realizar el clasificador aprenderán los principios de redes neuronales siguiendo este plan de trabajo:

1. Realicen el siguiente tutorial de cómo construir una red neuronal usando Keras https://www.youtube.com/playlist?list=PLZbbT5o_s2xrwRnXk_yCPtnqqo4_u2YGL
2. El tutorial de *Deeplizard* viene acompañado con una parte teórica, la cual complementa la parte práctica: https://www.youtube.com/playlist?list=PLZbbT5o_s2xq7LwI2y8_QtvuXZedL6tQU

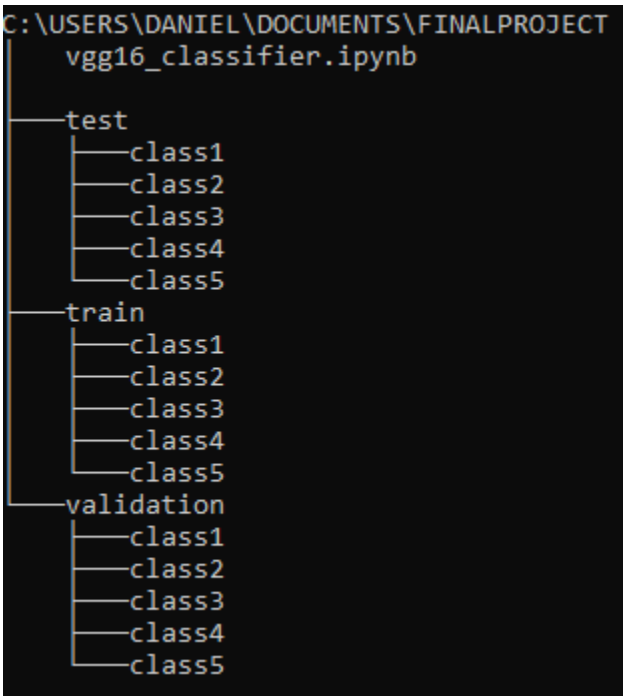
Recomendaciones

- De los videos teóricos de la serie “Machine Learning & Deep Learning Fundamentals”, pongan especial atención en los videos de CNN, backpropagation, learnable parameters y fine-tuning.
- Los videos del tutorial en *Deeplizard* fueron diseñados en el 2017, a la fecha han ocurrido cambios en la librería de Keras y Tensorflow, por lo que recomiendo revisen el blog de https://deeplizard.com/learn/playlist/PLZbbT5o_s2xrwRnXk_yCPtnqqo4_u2YGL el cual actualizan periódicamente.
- En caso de que tengan complicaciones instalando Keras, recomiendo utilicen Google colab el cual es una plataforma gratuita que ya tiene las librerías necesarias: https://youtu.be/vVe648dJOdI
  Nota: el setup de CUDA no es requisito por si desean trabajar localmente y no utilizar Google colab, por lo que pueden omitir el video #3 de la seria de Keras.
- Pueden utilizar las siguientes bases de datos de imágenes para descargar su set de datos con su web scrapper:

    - http://www.image-net.org/
    - https://pixabay.com/
    - https://www.pexels.com/license/
    - https://stocksnap.io/
    - https://unsplash.com/
- Para mejorar el desempeño de su modelo, experimenten con diferentes epochs en su etapa de entrenamiento y varíen la cantidad de capaz que están re-entrenando al tiempo de hacer la transferencia de aprendizaje (transfer learning). Un set de datos extenso favorecerá el desempeño, por lo que recomiendo hagan una inversión significativa en la recolección de imágenes y utilicen las técnicas de data augmentation.

Entregables:

- Enlace al repositorio de github con su respectivo README.
- Reporte de cuartilla y media explicando cómo recolectaron su set de datos y cómo opera su código.
- Suban a Canvas un .zip que contenga la siguiente estructura:



  En donde cada carpeta class# corresponde a una de las categorías que van a identificar y esté llena de las imágenes que utilizaron para su proyecto. Vgg16_classifier.py o Vgg16_classifier.ipynb es su red neuronal.
- Autoevaluación: Asignen una calificación del 1 al 5 que corresponda a su trabajo en el proyecto, adicionalmente indiquen quien fue la o el MVP del equipo, es decir el estudiante que más aportó al proyecto.