

# Tarea 1 - Taller de Deep Learning

Fecha de entrega: 08 / 10 / 2023

Puntaje máximo: 20

## Introducción:

El objetivo de esta tarea es evaluar su conocimiento sobre Deep Learning aplicado a un caso de uso real. En particular, vamos a evaluar la performance de sus modelos en una tarea de **clasificación de escenas**.

## Dataset:

El dataset a ser utilizado consiste de más de **13.000** imágenes de entrenamiento agrupadas en **6** posibles “escenarios”: **edificios, bosques, glaciares, montañas, océanos, calles**.

Pueden descargarlo en el siguiente [link](#).

El mismo contiene dos carpetas (**train\_set** y **test\_set**) donde las imágenes están agrupadas en sub carpetas de acuerdo a su categoría.

## Tarea:

Tienen total libertad sobre cómo implementar y resolver el problema, así como las técnicas y herramientas que quieran usar, recomendamos usar colab por simplicidad pero pueden implementarlo en sus máquinas si así lo prefieren. La única limitante es que esperamos que la entrega sea en formato **.ipynb** (Jupyter Notebook).

Se espera que implementen los modelos utilizados a mano y los entrenen específicamente en este dataset. Pueden implementar arquitecturas preexistentes como las que vimos en clase pero no están permitidos los modelos pre-entrenados.

Esperamos que las decisiones tomadas sobre el preprocesamiento de las imágenes (transforms, augmentation, etc.) sean resultado de la exploración del dataset y estén propiamente justificadas (una sección de exploración en el notebook con comentarios es suficiente).

Como se mencionó anteriormente, cuentan con un set de datos de test, el cual **no puede ser utilizado para entrenar**, su finalidad es reportar métricas de su solución.

En particular les pedimos que reporten: **accuracy, precision, recall y f1**, todas pueden ser obtenidas usando [sklearn](#).

También se espera poder observar la evolución del modelo (en los datos de train y validación) a medida que se entrena (logs, gráficas, etc).

## Guía para Colab:

1. Descargar el dataset y subirlo a google drive dentro de una carpeta “data”.
2. Ejecutar el siguiente código en la primera celda del notebook:

```
from google.colab import drive
drive.mount("/content/drive")

! cp "/content/drive/My Drive/data/scene_classification.zip" .
! unzip -q scene_classification.zip
! rm scene_classification.zip

# Luego de ejecutar esto deberían quedar 2 carpetas en la ruta de colab: train_set
# con las imágenes de entrenamiento y test_set con las de prueba.
```

3. Hacer click en el link, autorizar y pegar el link en el cuadro de texto. Esto debe hacerse cada vez que inicializamos el notebook, pero solo una vez por ejecución.
4. La ruta a la carpeta con imágenes de entrenamiento es simplemente “train\_set” y para test “test set”. Esto puede ser usado por **ImageFolder** de *torchvision* para leer las imágenes directamente.
5. Cargando el dataset

```
import torch
from torch.utils.data import DataLoader
from torchvision.datasets import ImageFolder

train_dataset = ImageFolder("train_set")
test_dataset = ImageFolder("test_set")

# Batch size de ejemplo.
BATCH_SIZE = 10

# Separamos en train y validation
train_size = int(0.8 * len(full_dataset))
valid_size = len(full_dataset) - train_size

train, validation = torch.utils.data.random_split(full_dataset, [train_size,
valid_size])

print(f"{len(train)} Training images, {len(validation)} Validation images")

# Podemos usar data loaders como vimos en el práctico.
train_loader = DataLoader(train, batch_size=BATCH_SIZE, shuffle=True,
pin_memory=True)
val_loader = DataLoader(validation, batch_size=BATCH_SIZE, pin_memory=True)
test_loader = DataLoader(test_dataset, batch_size=BATCH_SIZE, pin_memory=True)
```