Meeting 6.1 - Extra tasks - HDF5

Pirms šī sagatavot numpy mmap dataset!

Dataset:

https://www.kaggle.com/hsankesara/flickr-image-dataset

1. Implementēt data pre-processor script, izmantojot HDF5 formātu un pēc tam uztaisīt, ka torch.data.utils.Dataset to izmanto, lai **getitem** ielādētu no šī faila pointera
   https://docs.h5py.org/en/stable/

   Šim formātam nav nepieciešams blakus vēl viens fails, jo labels var glabāt kā dictionary key

2. Implementēt data pre-processor + dataset script, izmantojot cupy memory mapping

   https://github.com/cupy/cupy/issues/3431

   https://stackoverflow.com/questions/57752516/how-to-use-cuda-pinned-zero-copy-memory-for-a-memory-mapped-file

3. Pa tiešo file based-dataset (https://pytorch.org/tutorials/beginner/basics/data_tutorial.html)

```
import os
import pandas as pd
from torchvision.io import read_image

class CustomImageDataset(Dataset):
    def __init__(self, annotations_file, img_dir, transform=None, target_transform=None):
        self.img_labels = pd.read_csv(annotations_file)
        self.img_dir = img_dir
        self.transform = transform
        self.target_transform = target_transform

    def __len__(self):
        return len(self.img_labels)

    def __getitem__(self, idx):
        img_path = os.path.join(self.img_dir, self.img_labels.iloc[idx, 0])
        image = read_image(img_path)
        label = self.img_labels.iloc[idx, 1]
        if self.transform:
            image = self.transform(image)
        if self.target_transform:
            label = self.target_transform(label)
        return image, label
```

4. Notestēt apmācību uz 10 epochs, salīdzinot izpildes ātrumu ar `time.time()` mmap, hdf5, cupy mmap un file-based