Attendance Project

Juliana Varela

The aim of the project is to propose a DL model capable of **filling in the attendance sheet after taking a photo of the classroom.**

- Step 1: identify the faces in the image and thus the number of people in the room (segmentation problem),
- Step 2: recognize the gender of each face,
- Step 3: identify the person's identity (you'll need to describe exactly how to do this and what is the constraint).

```
!pip install ultralytics
```

```
Collecting ultralytics
  Downloading ultralytics-8.3.99-py3-none-any.whl.metadata (37 kB)
Requirement already satisfied: numpy<=2.1.1,>=1.23.0 in
/usr/local/lib/python3.11/dist-packages (from ultralytics) (2.0.2)
Requirement already satisfied: matplotlib>=3.3.0 in
/usr/local/lib/python3.11/dist-packages (from ultralytics) (3.10.0)
Requirement already satisfied: opencv-python>=4.6.0 in
/usr/local/lib/python3.11/dist-packages (from ultralytics) (4.11.0.86)
Requirement already satisfied: pillow>=7.1.2 in
/usr/local/lib/python3.11/dist-packages (from ultralytics) (11.1.0)
Requirement already satisfied: pyyaml>=5.3.1 in
/usr/local/lib/python3.11/dist-packages (from ultralytics) (6.0.2)
Requirement already satisfied: requests>=2.23.0 in
/usr/local/lib/python3.11/dist-packages (from ultralytics) (2.32.3)
Requirement already satisfied: scipy>=1.4.1 in
/usr/local/lib/python3.11/dist-packages (from ultralytics) (1.14.1)
Requirement already satisfied: torch>=1.8.0 in
/usr/local/lib/python3.11/dist-packages (from ultralytics)
(2.6.0+cu124)
Requirement already satisfied: torchvision>=0.9.0 in
/usr/local/lib/python3.11/dist-packages (from ultralytics)
(0.21.0+cu124)
Requirement already satisfied: tqdm>=4.64.0 in
/usr/local/lib/python3.11/dist-packages (from ultralytics) (4.67.1)
Requirement already satisfied: psutil in
/usr/local/lib/python3.11/dist-packages (from ultralytics) (5.9.5)
Requirement already satisfied: py-cpuinfo in
/usr/local/lib/python3.11/dist-packages (from ultralytics) (9.0.0)
Requirement already satisfied: pandas>=1.1.4 in
/usr/local/lib/python3.11/dist-packages (from ultralytics) (2.2.2)
Requirement already satisfied: seaborn>=0.11.0 in
```

```
/usr/local/lib/python3.11/dist-packages (from ultralytics) (0.13.2)
Collecting ultralytics-thop>=2.0.0 (from ultralytics)
  Downloading ultralytics_thop-2.0.14-py3-none-any.whl.metadata (9.4
kB)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib>=3.3.0-
>ultralytics) (1.3.1)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.11/dist-packages (from matplotlib>=3.3.0-
>ultralytics) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.11/dist-packages (from matplotlib>=3.3.0-
>ultralytics) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib>=3.3.0-
>ultralytics) (1.4.8)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.11/dist-packages (from matplotlib>=3.3.0-
>ultralytics) (24.2)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib>=3.3.0-
>ultralytics) (3.2.1)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.11/dist-packages (from matplotlib>=3.3.0-
>ultralytics) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.11/dist-packages (from pandas>=1.1.4-
>ultralytics) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.11/dist-packages (from pandas>=1.1.4-
>ultralytics) (2025.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.23.0-
>ultralytics) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.23.0-
>ultralytics) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.23.0-
>ultralytics) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.23.0-
>ultralytics) (2025.1.31)
Requirement already satisfied: filelock in
/usr/local/lib/python3.11/dist-packages (from torch>=1.8.0-
>ultralytics) (3.18.0)
Requirement already satisfied: typing-extensions>=4.10.0 in
/usr/local/lib/python3.11/dist-packages (from torch>=1.8.0-
>ultralytics) (4.12.2)
```

```
Requirement already satisfied: networkx in
/usr/local/lib/python3.11/dist-packages (from torch>=1.8.0-
>ultralytics) (3.4.2)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.11/dist-packages (from torch>=1.8.0-
>ultralytics) (3.1.6)
Requirement already satisfied: fsspec in
/usr/local/lib/python3.11/dist-packages (from torch>=1.8.0-
>ultralytics) (2025.3.0)
Collecting nvidia-cuda-nvrtc-cu12==12.4.127 (from torch>=1.8.0-
>ultralytics)
  Downloading nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cuda-runtime-cu12==12.4.127 (from torch>=1.8.0-
>ultralytics)
  Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cuda-cupti-cu12==12.4.127 (from torch>=1.8.0-
>ultralytics)
  Downloading nvidia_cuda_cupti_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cudnn-cu12==9.1.0.70 (from torch>=1.8.0-
>ultralytics)
  Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-
manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cublas-cu12==12.4.5.8 (from torch>=1.8.0-
>ultralytics)
  Downloading nvidia_cublas_cu12-12.4.5.8-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cufft-cu12==11.2.1.3 (from torch>=1.8.0-
>ultralytics)
  Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-curand-cu12==10.3.5.147 (from torch>=1.8.0-
>ultralytics)
  Downloading nvidia_curand_cu12-10.3.5.147-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cusolver-cu12==11.6.1.9 (from torch>=1.8.0-
>ultralytics)
  Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-
manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cusparse-cu12==12.3.1.170 (from torch>=1.8.0-
>ultralytics)
  Downloading nvidia_cusparse_cu12-12.3.1.170-py3-none-
manylinux2014_x86_64.whl.metadata (1.6 kB)
Requirement already satisfied: nvidia-cusparselt-cu12==0.6.2 in
/usr/local/lib/python3.11/dist-packages (from torch>=1.8.0-
>ultralytics) (0.6.2)
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in
```

```
/usr/local/lib/python3.11/dist-packages (from torch>=1.8.0->ultralytics) (2.21.5)
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages (from torch>=1.8.0->ultralytics) (12.4.127)
Collecting nvidia-nvjitlink-cu12==12.4.127 (from torch>=1.8.0->ultralytics)
  Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)
Requirement already satisfied: triton==3.2.0 in /usr/local/lib/python3.11/dist-packages (from torch>=1.8.0->ultralytics) (3.2.0)
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.11/dist-packages (from torch>=1.8.0->ultralytics) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from sympy==1.13.1->torch>=1.8.0->ultralytics) (1.3.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.7->matplotlib>=3.3.0->ultralytics) (1.17.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from jinja2->torch>=1.8.0->ultralytics) (3.0.2)
Downloading ultralytics-8.3.99-py3-none-any.whl (976 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 976.9/976.9 kB 58.4 MB/s eta 0:00:00
anylinux2014_x86_64.whl (363.4 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 363.4/363.4 MB 3.0 MB/s eta 0:00:00
anylinux2014_x86_64.whl (13.8 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.8/13.8 MB 115.1 MB/s eta 0:00:00
anylinux2014_x86_64.whl (24.6 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.6/24.6 MB 89.9 MB/s eta 0:00:00
e_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (883 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 883.7/883.7 kB 56.4 MB/s eta 0:00:00
anylinux2014_x86_64.whl (664.8 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 664.8/664.8 MB 2.2 MB/s eta 0:00:00
anylinux2014_x86_64.whl (211.5 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 211.5/211.5 MB 4.7 MB/s eta 0:00:00
anylinux2014_x86_64.whl (56.3 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.3/56.3 MB 40.1 MB/s eta 0:00:00
anylinux2014_x86_64.whl (127.9 MB)
```

```
 127.9/127.9 MB 18.6 MB/s eta
0:00:00
anylinux2014_x86_64.whl (207.5 MB)
 207.5/207.5 MB 4.4 MB/s eta
0:00:00
anylinux2014_x86_64.whl (21.1 MB)
 21.1/21.1 MB 95.7 MB/s eta
0:00:00
e-cu12, nvidia-cuda-nvrtc-cu12, nvidia-cuda-cupti-cu12, nvidia-cublas-
cu12, nvidia-cusparse-cu12, nvidia-cudnn-cu12, nvidia-cusolver-cu12,
ultralytics-thop, ultralytics
  Attempting uninstall: nvidia-nvjitlink-cu12
    Found existing installation: nvidia-nvjitlink-cu12 12.5.82
    Uninstalling nvidia-nvjitlink-cu12-12.5.82:
      Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
  Attempting uninstall: nvidia-curand-cu12
    Found existing installation: nvidia-curand-cu12 10.3.6.82
    Uninstalling nvidia-curand-cu12-10.3.6.82:
      Successfully uninstalled nvidia-curand-cu12-10.3.6.82
  Attempting uninstall: nvidia-cufft-cu12
    Found existing installation: nvidia-cufft-cu12 11.2.3.61
    Uninstalling nvidia-cufft-cu12-11.2.3.61:
      Successfully uninstalled nvidia-cufft-cu12-11.2.3.61
  Attempting uninstall: nvidia-cuda-runtime-cu12
    Found existing installation: nvidia-cuda-runtime-cu12 12.5.82
    Uninstalling nvidia-cuda-runtime-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-runtime-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-nvrtc-cu12
    Found existing installation: nvidia-cuda-nvrtc-cu12 12.5.82
    Uninstalling nvidia-cuda-nvrtc-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-nvrtc-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-cupti-cu12
    Found existing installation: nvidia-cuda-cupti-cu12 12.5.82
    Uninstalling nvidia-cuda-cupti-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-cupti-cu12-12.5.82
  Attempting uninstall: nvidia-cublas-cu12
    Found existing installation: nvidia-cublas-cu12 12.5.3.2
    Uninstalling nvidia-cublas-cu12-12.5.3.2:
      Successfully uninstalled nvidia-cublas-cu12-12.5.3.2
  Attempting uninstall: nvidia-cusparse-cu12
    Found existing installation: nvidia-cusparse-cu12 12.5.1.3
    Uninstalling nvidia-cusparse-cu12-12.5.1.3:
      Successfully uninstalled nvidia-cusparse-cu12-12.5.1.3
  Attempting uninstall: nvidia-cudnn-cu12
    Found existing installation: nvidia-cudnn-cu12 9.3.0.75
    Uninstalling nvidia-cudnn-cu12-9.3.0.75:
      Successfully uninstalled nvidia-cudnn-cu12-9.3.0.75
  Attempting uninstall: nvidia-cusolver-cu12
    Found existing installation: nvidia-cusolver-cu12 11.6.3.83
```

```
    Uninstalling nvidia-cusolver-cu12-11.6.3.83:
      Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83
Successfully installed nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-cupti-cu12-12.4.127 nvidia-cuda-nvrtc-cu12-12.4.127 nvidia-cuda-runtime-cu12-12.4.127 nvidia-cudnn-cu12-9.1.0.70 nvidia-cufft-cu12-11.2.1.3 nvidia-curand-cu12-10.3.5.147 nvidia-cusolver-cu12-11.6.1.9 nvidia-cusparse-cu12-12.3.1.170 nvidia-nvjitlink-cu12-12.4.127 ultralytics-8.3.99 ultralytics-thop-2.0.14
```

{"id":"bbb3c3725e324fcd84ce8884064c659e","pip_warning":{"packages":["nvidia"]}}

```python
import os
import re
import requests

import cv2
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt


import random
from PIL import Image
from torch.utils.data import Dataset, DataLoader
from collections import defaultdict

import torch
import torch.nn as nn
import torch.optim as optim
import torchvision.transforms as transforms
from torchvision.models import vit_b_16
from transformers import ViTFeatureExtractor, ViTForImageClassification
from ultralytics import YOLO


device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

DEEP LEARNING Approach

For this project the methodology will be the following:

- Step 1: In order to identify the faces and number of people we will use the model yolov8.
- Step 2: Then we will proceed to do a gender classification for these faces.

- Step 3: To identify the person's identity we will train a model of ViT with multiple individual images where the labels will be the name of each person.

Loading Data

```
#Finding the exact name of my folder
'''!ls -l /content/drive
!ls -l /content/drive/MyDrive/'''
```

{"type":"string"}

```
datasetRoot = 'Photos/'

# Printing all contents of main folder
if os.path.exists(datasetRoot):

  print("Shared folder contents:", os.listdir(datasetRoot))
  data = os.listdir(datasetRoot)
  # Getting the labeledPhotos Folder directory
  for item in data:
    if item == 'Labeled Photos':
      labeledPhotos = datasetRoot + '/' + item
else:
    print("Folder not found at:", datasetRoot)
```

```
Shared folder contents: ['Labeled Photos']
```

```
 # Get all of the images in the folder
def find_image_files(directory):

  image_files = []

  for root, dirs, files in os.walk(directory):
    for file in files:
      if file.lower().endswith(('.png', '.jpg', '.jpeg')):
        image_path = os.path.join(root, file)
        if "Média" not in file: #ignore files with Média in their name
because they are not of the classmates
          image_files.append(image_path)
  return image_files
```

Data preparation

Individual Images and labels

```
def extract_name_number(filename):
  # Remove extension and path
  basename = os.path.basename(filename)
  name_with_number = os.path.splitext(basename)[0]
```

```
  # We split the name from the numbers
  match = re.match(r"([a-zA-Z]+)(\d+)", name_with_number)
  if match:
    return match.group(1), int(match.group(2))
  return None, None

def data_augmentation(image):
  new_images = []

  # Flipping the image horizontally
  flipped = cv2.flip(image, 1)
  new_images.append(flipped)

  # This is from https://opencv.courses/blog/image-enhancement-with-
opencv/
  for alpha in [1.0, 1.5, 0.9]:
    for beta in [0, 20, -10]:
      brigtht = cv2.convertScaleAbs(image, alpha=1.5, beta= 20)

  new_images.append(brigtht)

  return new_images

def data_engineering(image_files):

  # We creata dict with defaultdict
  img_data = defaultdict(dict)
  final_data = []

  # We go through the images in the image files previously computed
and apply the extract_name_number function
  for img_path in image_files:

    name, number = extract_name_number(img_path)
    if name and number:
      # Read image, resize it and normaiwe it
      img = cv2.imread(img_path)
      img = cv2.resize(img, (224, 224))
      img = img / 255.

      if img is not None:
        # Create name(y) and number(X) into the dict
        img_data[name][number] = []
        # Append the image into it
        img_data[name][number].append(img)

        # Now apply our previous data_augmentation function into it
        augmented_images = data_augmentation(img)
        img_data[name][number].extend(augmented_images) # We add them
to our img_data
```

```
  # Now go over the img_data
  for name, numbers_dict in img_data.items():
    # This is for the test data, images that are different than the ones before
    sorted_numbers = sorted(numbers_dict.keys())
    # One image by number is added into X, y is the label (name)
    imgs_X = [numbers_dict[num][0] for num in sorted_numbers]
    final_data.append({'X': imgs_X,'y': name})

  return final_data

individual_image_files = find_image_files(labeledPhotos)
dat = data_engineering(individual_image_files)
```

```
Corrupt JPEG data: 2 extraneous bytes before marker 0xd7
Corrupt JPEG data: 1 extraneous bytes before marker 0xd3
Corrupt JPEG data: 1 extraneous bytes before marker 0xd0
Corrupt JPEG data: 1 extraneous bytes before marker 0xd5
Corrupt JPEG data: 1 extraneous bytes before marker 0xd0
```

```
# Class to use data and a tansformer to transform it
class FaceDataset(Dataset):
    def __init__(self, data, transform):
        self.data = data
        self.transform = transform

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

    def __getitem__(self, idx):
        sample = self.data[idx]
        # Images per label(student names)
        images = sample['X']
        label = sample['y']
        image = random.choice(images)

        # Make image (NumPy array) to PIL and transform it
        image = Image.fromarray((image * 255).astype(np.uint8))  # Convert back to uint8
        image = self.transform(image)

        return image, label

transform = transforms.Compose([
    transforms.ToTensor()
])
# transforming the data into tensors
dataset = FaceDataset(dat, transform)
```

We visualize the dataset images and their labels

```python
# Sample a few images and display them
fig, axes = plt.subplots(2, 3, figsize=(10, 6))
for i in range(6):
    id = random.randint(0, len(dataset)-1)
    image, label = dataset[id]
    # Convert tensor back to image
    image = image.permute(1, 2, 0).numpy()

    row = i // 3
    col = i % 3
    axes[row, col].imshow(image)
    axes[row, col].set_title(f"Label: {label}")
    axes[row, col].axis('off')
plt.tight_layout()
plt.show()
```

Group Images

```python
group_folders1 = datasetRoot + '/' + 'train'
group_folders2 = datasetRoot + '/' + 'From_V'
group_folders3 = datasetRoot + '/' + 'Group_Photo_Raw'

group_image_files = []
for i in range(1, 4):
  group_folders = locals()[f"group_folders{i}"]
```

```
  print(f"Searching for images in: {group_folders}")
  group_image_files.extend(find_image_files(group_folders))

print(f"Found {len(group_image_files)} image files.")
```

```
Searching for images in: /content/drive/.shortcut-targets-by-id/1-c-cpJ43qE4qG1aMoQtLzz75p8uBULWy/DATA ML DL project n°02 /train
Searching for images in: /content/drive/.shortcut-targets-by-id/1-c-cpJ43qE4qG1aMoQtLzz75p8uBULWy/DATA ML DL project n°02 /From_V
Searching for images in: /content/drive/.shortcut-targets-by-id/1-c-cpJ43qE4qG1aMoQtLzz75p8uBULWy/DATA ML DL project n°02 /Group_Photo_Raw
Found 82 image files.
```

Models

To do the face identification which will allow us to compare the people in the group pictures with their individual photos we will train a ViT (Visual image Transformer model)

```
labels = set()
for name in dat:
  labels.add(name['y'])
num_labels = len(labels)

#Face Classification Using ViT
feature_extractor = ViTFeatureExtractor.from_pretrained("google/vit-base-patch16-224-in21k")
viT_model = ViTForImageClassification.from_pretrained(
    "google/vit-base-patch16-224-in21k",
    num_labels=num_labels # Change based on the number of people
)
viT_model.to(device)
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
```

{"model_id":"ae9831794e9c4c96b0e5e9c93b197ca6","version_major":2,"version_minor":0}

```
/usr/local/lib/python3.11/dist-packages/transformers/models/vit/feature_extraction_vit.py:28: FutureWarning: The class ViTFeatureExtractor is deprecated and will be removed in version 5 of
```

```
Transformers. Please use ViTImageProcessor instead.
  warnings.warn(
```

{"model_id":"5ff5c405119948ecb8a31db64d15535d","version_major":2,"vers
ion_minor":0}

{"model_id":"4907ce0d7ca94a029801a622a076098e","version_major":2,"vers
ion_minor":0}

```
Some weights of ViTForImageClassification were not initialized from
the model checkpoint at google/vit-base-patch16-224-in21k and are
newly initialized: ['classifier.bias', 'classifier.weight']
You should probably TRAIN this model on a down-stream task to be able
to use it for predictions and inference.
```

```
ViTForImageClassification(
  (vit): ViTModel(
    (embeddings): ViTEmbeddings(
      (patch_embeddings): ViTPatchEmbeddings(
        (projection): Conv2d(3, 768, kernel_size=(16, 16), stride=(16,
16))
      )
      (dropout): Dropout(p=0.0, inplace=False)
    )
    (encoder): ViTEncoder(
      (layer): ModuleList(
        (0-11): 12 x ViTLayer(
          (attention): ViTAttention(
            (attention): ViTSelfAttention(
              (query): Linear(in_features=768, out_features=768,
bias=True)
              (key): Linear(in_features=768, out_features=768,
bias=True)
              (value): Linear(in_features=768, out_features=768,
bias=True)
            )
            (output): ViTSelfOutput(
              (dense): Linear(in_features=768, out_features=768,
bias=True)
              (dropout): Dropout(p=0.0, inplace=False)
            )
          )
          (intermediate): ViTIntermediate(
            (dense): Linear(in_features=768, out_features=3072,
bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): ViTOutput(
            (dense): Linear(in_features=3072, out_features=768,
bias=True)
```

```
            (dropout): Dropout(p=0.0, inplace=False)
          )
          (layernorm_before): LayerNorm((768,), eps=1e-12,
elementwise_affine=True)
          (layernorm_after): LayerNorm((768,), eps=1e-12,
elementwise_affine=True)
        )
      )
    )
    (layernorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
  )
  (classifier): Linear(in_features=768, out_features=17, bias=True)
)
```

Training the viT model

```
# We were having some issues with bt-labels variable because its not
encoded so we used this to encode them
label_mapping = {name: idx for idx, name in enumerate(set(labl['y']
for labl in dat))}

train_loader = DataLoader(dataset, batch_size=16, shuffle=True)

# Creating the optimizer and the loss function
optimizer = optim.Adam(viT_model.parameters(), lr=0.0001)
criterion = nn.CrossEntropyLoss() # Since its a multiclassificcation
problem

num_epochs = 5
for epoch in range(num_epochs):
  viT_model.train()
  running_loss = 0.0
  correct, total = 0, 0

  for images, bt_labels in train_loader:
    #pass them to device
    images = images.to(device)
    print(bt_labels)
    bt_labels = torch.tensor([label_mapping[label] for label in
bt_labels]).to(device)

    optimizer.zero_grad()

    outputs = viT_model(images).logits
    loss = criterion(outputs, bt_labels)
    loss.backward()
    optimizer.step()

    running_loss += loss.item()
    #print(running_loss)
```

```
    _, predicted = torch.max(outputs, 1)
    # We get the number of all of the correct predictions
    correct += (predicted == bt_labels).sum().item()
    total += bt_labels.size(0)

  accuracy = 100 * correct / total
```

```
('Ubaid', 'Ismail', 'Tuan', 'Youssef', 'Nestor', 'Melissa', 'Usman',
'Arion', 'Algassimou', 'Zeev', 'Gerard', 'Hadi', 'Hassan', 'Hasham',
'Juliana', 'Abhinav')
('Zahid',)
('Melissa', 'Ismail', 'Tuan', 'Nestor', 'Hasham', 'Gerard', 'Zeev',
'Abhinav', 'Hadi', 'Usman', 'Algassimou', 'Hassan', 'Ubaid', 'Zahid',
'Arion', 'Youssef')
('Juliana',)
('Hassan', 'Youssef', 'Gerard', 'Usman', 'Tuan', 'Zahid', 'Ubaid',
'Juliana', 'Hasham', 'Melissa', 'Ismail', 'Zeev', 'Hadi', 'Abhinav',
'Arion', 'Algassimou')
('Nestor',)
('Zahid', 'Ubaid', 'Usman', 'Juliana', 'Arion', 'Hassan', 'Gerard',
'Hadi', 'Abhinav', 'Algassimou', 'Youssef', 'Tuan', 'Ismail',
'Nestor', 'Hasham', 'Zeev')
('Melissa',)
('Zahid', 'Hasham', 'Hadi', 'Usman', 'Arion', 'Algassimou', 'Abhinav',
'Hassan', 'Zeev', 'Ismail', 'Nestor', 'Melissa', 'Youssef', 'Juliana',
'Gerard', 'Tuan')
('Ubaid',)
```

```
correct, accuracy
```

```
(9, 52.94117647058823)
```

Step 1: Using YOLO for object detection

In order to detect the faces of the people in a group image we will use the model yolov8 which is great for object detection

```
yolo = YOLO('yolov8n.pt')
```

```
Downloading
https://github.com/ultralytics/assets/releases/download/v8.3.0/yolov8n
.pt to 'yolov8n.pt'...

100%|██████████| 6.25M/6.25M [00:00<00:00, 376MB/s]
```

```
def detect_faces(group_image):
  #We pass the group images to our yolo model
  results = yolo(group_image)
  faces = []
```

```python
  # Here we will get the bounding boxes of the detected objects
  for box, clas in zip(results[0].boxes.xyxy, results[0].boxes.cls):
    # 0 class is for people only to makes sure we only detect them
    if clas ==0:
      # Get coordinates
      x1, y1, x2, y2 = map(int, box[:4])
      # Getting the faces region
      faces.append(group_image[y1:y2, x1:x2])

  return faces
```

We apply our detecting function to our actual group image files

```python
detected_faces = []

for image_path in group_image_files:
  image = cv2.imread(image_path)
  faces = detect_faces(image)
  detected_faces.extend(faces)
print(f'Total Detected faces:{len(detected_faces)}')
```

```
0: 640x640 15 persons, 1 bottle, 1 chair, 4 laptops, 8.0ms
Speed: 4.2ms preprocess, 8.0ms inference, 371.3ms postprocess per
image at shape (1, 3, 640, 640)

0: 640x640 12 persons, 2 backpacks, 2 handbags, 4 chairs, 1 dining
table, 2 laptops, 8.0ms
Speed: 4.0ms preprocess, 8.0ms inference, 1.4ms postprocess per image
at shape (1, 3, 640, 640)

0: 640x640 14 persons, 1 backpack, 1 handbag, 1 bottle, 4 chairs, 1
dining table, 2 laptops, 8.1ms
Speed: 3.3ms preprocess, 8.1ms inference, 1.5ms postprocess per image
at shape (1, 3, 640, 640)

0: 640x640 10 persons, 1 backpack, 1 handbag, 1 tie, 3 chairs, 1 tv,
8.0ms
Speed: 3.5ms preprocess, 8.0ms inference, 1.4ms postprocess per image
at shape (1, 3, 640, 640)

0: 640x640 10 persons, 1 tie, 9 chairs, 2 dining tables, 7 laptops, 1
book, 8.0ms
Speed: 2.8ms preprocess, 8.0ms inference, 1.4ms postprocess per image
at shape (1, 3, 640, 640)

0: 640x640 15 persons, 1 chair, 1 dining table, 5 laptops, 8.0ms
Speed: 3.7ms preprocess, 8.0ms inference, 1.4ms postprocess per image
at shape (1, 3, 640, 640)
```

```
0: 640x640 12 persons, 2 backpacks, 1 handbag, 1 cup, 5 chairs, 1 laptop, 8.0ms
Speed: 3.0ms preprocess, 8.0ms inference, 1.3ms postprocess per image at shape (1, 3, 640, 640)

0: 256x640 14 persons, 5 chairs, 2 laptops, 48.9ms
Speed: 2.0ms preprocess, 48.9ms inference, 1.5ms postprocess per image at shape (1, 3, 256, 640)

0: 256x640 13 persons, 5 chairs, 2 laptops, 7.3ms
Speed: 2.1ms preprocess, 7.3ms inference, 1.4ms postprocess per image at shape (1, 3, 256, 640)

0: 256x640 15 persons, 1 bottle, 6 chairs, 2 laptops, 7.6ms
Speed: 2.0ms preprocess, 7.6ms inference, 1.5ms postprocess per image at shape (1, 3, 256, 640)

0: 256x640 19 persons, 6 chairs, 4 laptops, 8.0ms
Speed: 2.1ms preprocess, 8.0ms inference, 1.7ms postprocess per image at shape (1, 3, 256, 640)

0: 256x640 17 persons, 1 cup, 7 chairs, 5 laptops, 6.9ms
Speed: 2.0ms preprocess, 6.9ms inference, 1.4ms postprocess per image at shape (1, 3, 256, 640)

0: 288x640 19 persons, 7 chairs, 2 laptops, 38.8ms
Speed: 2.0ms preprocess, 38.8ms inference, 1.3ms postprocess per image at shape (1, 3, 288, 640)

0: 320x640 17 persons, 1 bottle, 6 chairs, 6 laptops, 39.7ms
Speed: 2.2ms preprocess, 39.7ms inference, 1.3ms postprocess per image at shape (1, 3, 320, 640)

0: 256x640 10 persons, 1 bottle, 1 cup, 6 chairs, 1 dining table, 3 laptops, 7.8ms
Speed: 2.0ms preprocess, 7.8ms inference, 1.5ms postprocess per image at shape (1, 3, 256, 640)

0: 320x640 14 persons, 2 chairs, 2 laptops, 7.9ms
Speed: 2.2ms preprocess, 7.9ms inference, 1.3ms postprocess per image at shape (1, 3, 320, 640)

0: 256x640 13 persons, 2 ties, 4 laptops, 7.8ms
Speed: 2.0ms preprocess, 7.8ms inference, 1.3ms postprocess per image at shape (1, 3, 256, 640)

0: 288x640 19 persons, 4 chairs, 1 dining table, 5 laptops, 7.6ms
Speed: 1.9ms preprocess, 7.6ms inference, 1.4ms postprocess per image at shape (1, 3, 288, 640)
```

```
0: 224x640 17 persons, 1 chair, 2 laptops, 39.9ms
Speed: 1.7ms preprocess, 39.9ms inference, 1.4ms postprocess per image
at shape (1, 3, 224, 640)

0: 416x640 8 persons, 48.8ms
Speed: 2.4ms preprocess, 48.8ms inference, 1.4ms postprocess per image
at shape (1, 3, 416, 640)

0: 224x640 16 persons, 3 chairs, 7.4ms
Speed: 1.6ms preprocess, 7.4ms inference, 1.3ms postprocess per image
at shape (1, 3, 224, 640)

0: 192x640 15 persons, 2 chairs, 1 laptop, 39.0ms
Speed: 1.5ms preprocess, 39.0ms inference, 1.4ms postprocess per image
at shape (1, 3, 192, 640)

0: 160x640 15 persons, 43.0ms
Speed: 1.4ms preprocess, 43.0ms inference, 1.3ms postprocess per image
at shape (1, 3, 160, 640)

0: 320x640 6 persons, 7.6ms
Speed: 2.0ms preprocess, 7.6ms inference, 1.2ms postprocess per image
at shape (1, 3, 320, 640)

0: 256x640 11 persons, 1 bottle, 1 chair, 8.2ms
Speed: 1.9ms preprocess, 8.2ms inference, 1.3ms postprocess per image
at shape (1, 3, 256, 640)

0: 224x640 19 persons, 1 laptop, 7.9ms
Speed: 1.8ms preprocess, 7.9ms inference, 1.3ms postprocess per image
at shape (1, 3, 224, 640)

0: 224x640 16 persons, 6.9ms
Speed: 1.8ms preprocess, 6.9ms inference, 1.4ms postprocess per image
at shape (1, 3, 224, 640)

0: 384x640 8 persons, 1 bottle, 2 chairs, 45.3ms
Speed: 2.5ms preprocess, 45.3ms inference, 1.4ms postprocess per image
at shape (1, 3, 384, 640)

0: 256x640 11 persons, 7 chairs, 4 laptops, 7.8ms
Speed: 2.0ms preprocess, 7.8ms inference, 1.3ms postprocess per image
at shape (1, 3, 256, 640)

0: 448x640 14 persons, 2 backpacks, 1 cup, 5 chairs, 1 laptop, 40.2ms
Speed: 3.4ms preprocess, 40.2ms inference, 1.4ms postprocess per image
at shape (1, 3, 448, 640)

0: 448x640 15 persons, 3 chairs, 10 laptops, 6.9ms
Speed: 3.7ms preprocess, 6.9ms inference, 1.3ms postprocess per image
```

```
at shape (1, 3, 448, 640)

0: 448x640 10 persons, 1 tie, 1 bottle, 1 cup, 2 chairs, 1 tv, 1
laptop, 7.0ms
Speed: 3.6ms preprocess, 7.0ms inference, 1.4ms postprocess per image
at shape (1, 3, 448, 640)

0: 448x640 15 persons, 1 chair, 2 laptops, 7.4ms
Speed: 3.6ms preprocess, 7.4ms inference, 1.4ms postprocess per image
at shape (1, 3, 448, 640)

0: 448x640 10 persons, 1 tie, 1 cup, 9 chairs, 2 dining tables, 5
laptops, 6.9ms
Speed: 3.7ms preprocess, 6.9ms inference, 1.4ms postprocess per image
at shape (1, 3, 448, 640)

0: 448x640 14 persons, 7 chairs, 1 dining table, 6 laptops, 8.6ms
Speed: 4.1ms preprocess, 8.6ms inference, 1.4ms postprocess per image
at shape (1, 3, 448, 640)

0: 448x640 16 persons, 7 chairs, 6 laptops, 7.9ms
Speed: 3.7ms preprocess, 7.9ms inference, 1.6ms postprocess per image
at shape (1, 3, 448, 640)

0: 448x640 11 persons, 1 backpack, 1 handbag, 1 tie, 1 bottle, 1 cup,
2 chairs, 1 laptop, 7.0ms
Speed: 3.5ms preprocess, 7.0ms inference, 1.3ms postprocess per image
at shape (1, 3, 448, 640)

0: 448x640 18 persons, 1 handbag, 1 bottle, 7 chairs, 6 laptops, 7.2ms
Speed: 3.7ms preprocess, 7.2ms inference, 1.4ms postprocess per image
at shape (1, 3, 448, 640)

0: 480x640 16 persons, 1 handbag, 2 bottles, 7 chairs, 2 dining
tables, 1 laptop, 40.3ms
Speed: 3.7ms preprocess, 40.3ms inference, 1.4ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 3 persons, 1 backpack, 1 handbag, 1 chair, 1 dining table,
1 laptop, 6.9ms
Speed: 3.4ms preprocess, 6.9ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 3 persons, 1 backpack, 1 handbag, 3 chairs, 1 laptop, 7.1ms
Speed: 3.4ms preprocess, 7.1ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 11 persons, 1 cup, 11 chairs, 5 dining tables, 7 laptops, 1
cell phone, 6.8ms
Speed: 3.6ms preprocess, 6.8ms inference, 1.3ms postprocess per image
```

```
at shape (1, 3, 480, 640)

0: 480x640 12 persons, 2 cups, 12 chairs, 4 dining tables, 6 laptops,
1 cell phone, 7.0ms
Speed: 3.7ms preprocess, 7.0ms inference, 1.4ms postprocess per image
at shape (1, 3, 480, 640)

0: 640x480 3 persons, 1 cup, 2 chairs, 1 dining table, 3 laptops,
41.7ms
Speed: 4.0ms preprocess, 41.7ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 480x640 8 persons, 1 bottle, 1 cup, 9 chairs, 2 dining tables, 7
laptops, 7.6ms
Speed: 4.0ms preprocess, 7.6ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 1 cup, 2 chairs, 2 dining tables, 1 laptop, 1
cell phone, 1 book, 6.8ms
Speed: 3.7ms preprocess, 6.8ms inference, 1.4ms postprocess per image
at shape (1, 3, 480, 640)

0: 640x480 3 persons, 2 chairs, 1 laptop, 7.9ms
Speed: 3.6ms preprocess, 7.9ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 480x640 12 persons, 12 chairs, 2 dining tables, 5 laptops, 7.8ms
Speed: 3.6ms preprocess, 7.8ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 640x480 2 persons, 1 cup, 2 chairs, 2 laptops, 1 mouse, 1 keyboard,
9.4ms
Speed: 3.7ms preprocess, 9.4ms inference, 1.4ms postprocess per image
at shape (1, 3, 640, 480)

0: 480x640 9 persons, 1 bottle, 1 cup, 10 chairs, 3 dining tables, 4
laptops, 8.4ms
Speed: 3.6ms preprocess, 8.4ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 640x480 3 persons, 1 cup, 2 chairs, 1 dining table, 3 laptops, 3
cell phones, 7.5ms
Speed: 3.6ms preprocess, 7.5ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 2 persons, 1 cup, 2 chairs, 2 dining tables, 3 laptops, 1
cell phone, 6.9ms
Speed: 3.7ms preprocess, 6.9ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)
```

```
0: 640x480 2 persons, 1 cup, 1 chair, 4 laptops, 1 cell phone, 7.2ms
Speed: 3.8ms preprocess, 7.2ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 480x640 14 persons, 1 cup, 9 chairs, 2 dining tables, 4 laptops,
8.0ms
Speed: 3.8ms preprocess, 8.0ms inference, 1.4ms postprocess per image
at shape (1, 3, 480, 640)

0: 640x480 3 persons, 1 cup, 3 chairs, 1 dining table, 2 laptops, 1
cell phone, 7.5ms
Speed: 3.5ms preprocess, 7.5ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 2 persons, 1 cup, 2 chairs, 1 dining table, 3 laptops, 1
mouse, 1 keyboard, 6.8ms
Speed: 4.1ms preprocess, 6.8ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 3 persons, 1 skateboard, 3 chairs, 2 laptops, 6.8ms
Speed: 3.6ms preprocess, 6.8ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 3 persons, 2 chairs, 6.9ms
Speed: 3.7ms preprocess, 6.9ms inference, 1.4ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 4 persons, 3 chairs, 1 laptop, 7.0ms
Speed: 3.7ms preprocess, 7.0ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 3 persons, 1 cup, 1 laptop, 6.9ms
Speed: 3.7ms preprocess, 6.9ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 2 persons, 1 cup, 2 chairs, 1 dining table, 4 laptops, 1
mouse, 6.8ms
Speed: 3.7ms preprocess, 6.8ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 3 persons, 6.8ms
Speed: 3.7ms preprocess, 6.8ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 4 persons, 7.0ms
Speed: 4.1ms preprocess, 7.0ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 480x640 4 persons, 1 dining table, 4 laptops, 1 cell phone, 7.5ms
Speed: 3.7ms preprocess, 7.5ms inference, 1.3ms postprocess per image
```

```
at shape (1, 3, 480, 640)

0: 480x640 6 persons, 1 cup, 1 chair, 1 dining table, 1 laptop, 1 cell
phone, 7.3ms
Speed: 4.1ms preprocess, 7.3ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 3 persons, 1 dog, 2 chairs, 1 dining table, 1 laptop, 7.2ms
Speed: 3.6ms preprocess, 7.2ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 2 chairs, 1 dining table, 1 laptop, 7.3ms
Speed: 4.0ms preprocess, 7.3ms inference, 1.4ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 1 cup, 2 chairs, 2 dining tables, 1 laptop, 1
cell phone, 6.8ms
Speed: 3.7ms preprocess, 6.8ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 11 persons, 1 bottle, 1 cup, 7 chairs, 6 dining tables, 7
laptops, 1 cell phone, 1 book, 6.8ms
Speed: 3.5ms preprocess, 6.8ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 1 backpack, 1 handbag, 2 chairs, 1 dining table,
1 laptop, 7.1ms
Speed: 3.9ms preprocess, 7.1ms inference, 1.4ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 11 persons, 1 bottle, 1 cup, 8 chairs, 3 dining tables, 8
laptops, 6.9ms
Speed: 3.7ms preprocess, 6.9ms inference, 1.4ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 1 cup, 2 chairs, 3 dining tables, 1 laptop, 1
cell phone, 1 book, 6.8ms
Speed: 3.7ms preprocess, 6.8ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 1 cup, 2 chairs, 2 dining tables, 1 laptop, 2
cell phones, 7.0ms
Speed: 3.7ms preprocess, 7.0ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 1 handbag, 1 chair, 1 dining table, 1 laptop,
6.9ms
Speed: 3.7ms preprocess, 6.9ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)
```

```
0: 480x640 3 persons, 3 chairs, 1 dining table, 1 laptop, 1 cell
phone, 7.1ms
Speed: 3.9ms preprocess, 7.1ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 3 persons, 1 backpack, 2 chairs, 1 laptop, 7.1ms
Speed: 3.7ms preprocess, 7.1ms inference, 1.4ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 3 persons, 1 backpack, 1 chair, 1 dining table, 1 laptop,
7.4ms
Speed: 3.8ms preprocess, 7.4ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 3 persons, 2 chairs, 1 dining table, 1 laptop, 1 keyboard,
1 cell phone, 7.1ms
Speed: 3.7ms preprocess, 7.1ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 2 laptops, 1 keyboard, 2 cell phones, 7.1ms
Speed: 3.6ms preprocess, 7.1ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 1 bottle, 2 chairs, 2 laptops, 2 cell phones,
8.2ms
Speed: 4.1ms preprocess, 8.2ms inference, 1.9ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 1 couch, 3 laptops, 2 keyboards, 1 cell phone,
8.4ms
Speed: 3.9ms preprocess, 8.4ms inference, 1.6ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 7 persons, 1 bottle, 3 chairs, 2 dining tables, 5 laptops,
1 keyboard, 1 cell phone, 6.9ms
Speed: 3.4ms preprocess, 6.9ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)
Total Detected faces:747
```

We visualize it with a single picture

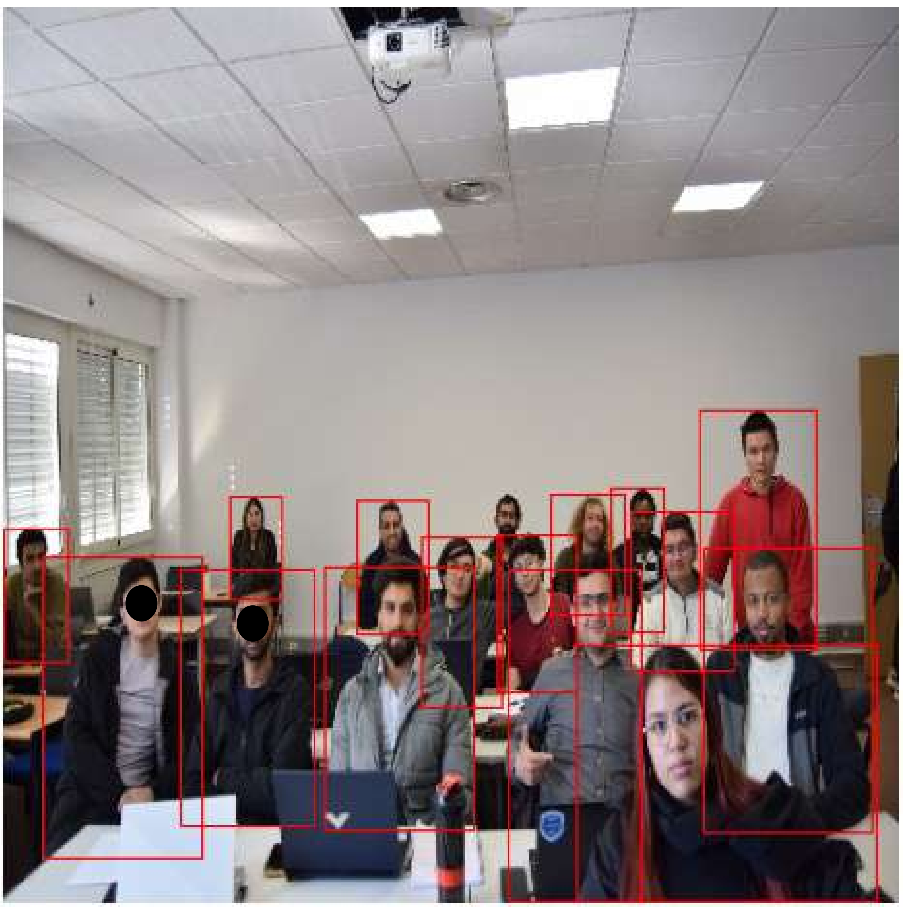
```
#Chose and read a group image
chosen_image = cv2.imread(group_image_files[0])
chosen_image_rgb = cv2.cvtColor(chosen_image, cv2.COLOR_BGR2RGB)
results = yolo(chosen_image)
# Show the bounding boxes
for box, clas in zip(results[0].boxes.xyxy, results[0].boxes.cls):
  if clas == 0: # to only see people
    x1, y1, x2, y2 = map(int, box[:4])
    cv2.rectangle(chosen_image_rgb, (x1, y1), (x2, y2), (255, 0, 0),
```

```
1)

plt.figure(figsize=(8, 6))
plt.imshow(chosen_image_rgb)
plt.axis("off")
plt.title('Detected Faces')
plt.show()


0: 640x640 15 persons, 1 bottle, 1 chair, 4 laptops, 8.7ms
Speed: 3.8ms preprocess, 8.7ms inference, 1.4ms postprocess per image
at shape (1, 3, 640, 640)
```

Detected Faces



Step 2: Gender Classification

We will use a pretrained gender classification model for this:
https://huggingface.co/prithivMLmods/Gender-Classifier-Mini

```
!pip install -q transformers torch pillow
```

```python
# this code comes from https://huggingface.co/prithivMLmods/Gender-
Classifier-Mini
from transformers import AutoImageProcessor
from transformers import SiglipForImageClassification
from transformers.image_utils import load_image

# Load model and processor
model_name = "prithivMLmods/Gender-Classifier-Mini"
model = SiglipForImageClassification.from_pretrained(model_name)
processor = AutoImageProcessor.from_pretrained(model_name)

def gender_classification(image):
    """Predicts gender category for an image."""
    image = Image.fromarray(image).convert("RGB")
    inputs = processor(images=image, return_tensors="pt")

    with torch.no_grad():
        outputs = model(**inputs)
        logits = outputs.logits
        probs = torch.nn.functional.softmax(logits,
dim=1).squeeze().tolist()

    labels = {"0": "Female ♀", "1": "Male ♂"}
    predictions = {labels[str(i)]: round(probs[i], 3) for i in
range(len(probs))}

    return predictions
```

{"model_id":"dcab95ae7aa448b7b11eba698cf497bf","version_major":2,"version_minor":0}

{"model_id":"1ad1e88895b54ace936aa82e2d1a1239","version_major":2,"version_minor":0}

{"model_id":"625675f3287148efb5637d02ff50031e","version_major":2,"version_minor":0}

```
Using a slow image processor as `use_fast` is unset and a slow
processor was saved with this model. `use_fast=True` will be the
default behavior in v4.50, even if the model was saved with a slow
processor. This will result in minor differences in outputs. You'll
still be able to use a slow processor with `use_fast=False`.
```

Now we will apply the gender classification to the detected faces by yolo in the group pictures

We visualize the gender classification with some samples

```python
for i in range(7):
  gender_proba = gender_classification(detected_faces[i])
  predicted_gender = max(gender_proba, key=gender_proba.get)
  probability = gender_proba[predicted_gender]
```

```python
# Display the image with bounding box and gender label
plt.figure(figsize=(4, 4))
plt.imshow(cv2.cvtColor(detected_faces[i], cv2.COLOR_BGR2RGB))
plt.title(f"Gender: {predicted_gender}")
plt.axis('off')
plt.show()
```

Gender: Male ♂

Gender: Male ♂

Gender: Male ♂

Gender: Male ♂

Gender: Male ♂

Gender: Female ♀

Gender: Male ♂

Step 3: Using the trained ViT model and YOLO

```
def get_names(faces):
  names = []
  for face in faces:
    # Preprocess each invididual face image
    face_pic = Image.fromarray(cv2.cvtColor(face, cv2.COLOR_BGR2RGB))
    prep = feature_extractor(images=face_pic, 
return_tensors="pt").to(device) #pt means type of tensor
    # Predictions from ViT
    with torch.no_grad():
      outputs = viT_model(**prep)
      logits = outputs.logits # get the scores
      # We take the name with the highest probability and predict 
thats the name
      predicted_class_id = logits.argmax(-1).item()
      #We do the oposite of before and get the actual name label based 
on the encoded label
      predicted_label = list(label_mapping.keys())
[list(label_mapping.values()).index(predicted_class_id)]
      names.append(predicted_label)
  return names

group_labels = []
final_data = []

for image_path in group_image_files:
    image = cv2.imread(image_path)
    faces = detect_faces(image)
    group_labels = get_names(faces)
```

```
    group_labels.append(group_labels)
    group_image_file_name = image_path.replace(datasetRoot,
'').lstrip(os.sep)
    final_data.append({'group_image_file_name': group_image_file_name,
'student_names': group_labels})
```

```
0: 640x640 15 persons, 1 bottle, 1 chair, 4 laptops, 8.3ms
Speed: 3.2ms preprocess, 8.3ms inference, 1.6ms postprocess per image
at shape (1, 3, 640, 640)

0: 640x640 12 persons, 2 backpacks, 2 handbags, 4 chairs, 1 dining
table, 2 laptops, 6.7ms
Speed: 2.8ms preprocess, 6.7ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 640)

0: 640x640 14 persons, 1 backpack, 1 handbag, 1 bottle, 4 chairs, 1
dining table, 2 laptops, 7.7ms
Speed: 2.8ms preprocess, 7.7ms inference, 1.5ms postprocess per image
at shape (1, 3, 640, 640)

0: 640x640 10 persons, 1 backpack, 1 handbag, 1 tie, 3 chairs, 1 tv,
9.0ms
Speed: 4.0ms preprocess, 9.0ms inference, 1.8ms postprocess per image
at shape (1, 3, 640, 640)

0: 640x640 10 persons, 1 tie, 9 chairs, 2 dining tables, 7 laptops, 1
book, 8.1ms
Speed: 3.8ms preprocess, 8.1ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 640)

0: 640x640 15 persons, 1 chair, 1 dining table, 5 laptops, 10.3ms
Speed: 4.0ms preprocess, 10.3ms inference, 1.7ms postprocess per image
at shape (1, 3, 640, 640)

0: 640x640 12 persons, 2 backpacks, 1 handbag, 1 cup, 5 chairs, 1
laptop, 10.0ms
Speed: 3.8ms preprocess, 10.0ms inference, 1.8ms postprocess per image
at shape (1, 3, 640, 640)

0: 256x640 14 persons, 5 chairs, 2 laptops, 14.0ms
Speed: 2.8ms preprocess, 14.0ms inference, 1.9ms postprocess per image
at shape (1, 3, 256, 640)

0: 256x640 13 persons, 5 chairs, 2 laptops, 9.8ms
Speed: 2.7ms preprocess, 9.8ms inference, 1.4ms postprocess per image
at shape (1, 3, 256, 640)

0: 256x640 15 persons, 1 bottle, 6 chairs, 2 laptops, 9.1ms
Speed: 2.5ms preprocess, 9.1ms inference, 1.7ms postprocess per image
```

```
at shape (1, 3, 256, 640)

0: 256x640 19 persons, 6 chairs, 4 laptops, 7.2ms
Speed: 2.6ms preprocess, 7.2ms inference, 1.4ms postprocess per image
at shape (1, 3, 256, 640)

0: 256x640 17 persons, 1 cup, 7 chairs, 5 laptops, 8.7ms
Speed: 1.9ms preprocess, 8.7ms inference, 1.4ms postprocess per image
at shape (1, 3, 256, 640)

0: 288x640 19 persons, 7 chairs, 2 laptops, 7.2ms
Speed: 2.0ms preprocess, 7.2ms inference, 1.2ms postprocess per image
at shape (1, 3, 288, 640)

0: 320x640 17 persons, 1 bottle, 6 chairs, 6 laptops, 8.1ms
Speed: 2.6ms preprocess, 8.1ms inference, 1.7ms postprocess per image
at shape (1, 3, 320, 640)

0: 256x640 10 persons, 1 bottle, 1 cup, 6 chairs, 1 dining table, 3
laptops, 8.7ms
Speed: 1.8ms preprocess, 8.7ms inference, 1.5ms postprocess per image
at shape (1, 3, 256, 640)

0: 320x640 14 persons, 2 chairs, 2 laptops, 10.9ms
Speed: 2.2ms preprocess, 10.9ms inference, 1.6ms postprocess per image
at shape (1, 3, 320, 640)

0: 256x640 13 persons, 2 ties, 4 laptops, 7.7ms
Speed: 1.9ms preprocess, 7.7ms inference, 1.3ms postprocess per image
at shape (1, 3, 256, 640)

0: 288x640 19 persons, 4 chairs, 1 dining table, 5 laptops, 8.2ms
Speed: 2.0ms preprocess, 8.2ms inference, 1.7ms postprocess per image
at shape (1, 3, 288, 640)

0: 224x640 17 persons, 1 chair, 2 laptops, 7.5ms
Speed: 1.6ms preprocess, 7.5ms inference, 1.3ms postprocess per image
at shape (1, 3, 224, 640)

0: 416x640 8 persons, 9.7ms
Speed: 2.9ms preprocess, 9.7ms inference, 1.3ms postprocess per image
at shape (1, 3, 416, 640)

0: 224x640 16 persons, 3 chairs, 7.7ms
Speed: 1.8ms preprocess, 7.7ms inference, 1.3ms postprocess per image
at shape (1, 3, 224, 640)

0: 192x640 15 persons, 2 chairs, 1 laptop, 7.4ms
Speed: 1.5ms preprocess, 7.4ms inference, 1.3ms postprocess per image
at shape (1, 3, 192, 640)
```

```
0: 160x640 15 persons, 7.2ms
Speed: 1.3ms preprocess, 7.2ms inference, 1.3ms postprocess per image
at shape (1, 3, 160, 640)

0: 320x640 6 persons, 9.1ms
Speed: 1.8ms preprocess, 9.1ms inference, 1.3ms postprocess per image
at shape (1, 3, 320, 640)

0: 256x640 11 persons, 1 bottle, 1 chair, 7.3ms
Speed: 1.6ms preprocess, 7.3ms inference, 1.3ms postprocess per image
at shape (1, 3, 256, 640)

0: 224x640 19 persons, 1 laptop, 7.5ms
Speed: 2.0ms preprocess, 7.5ms inference, 1.3ms postprocess per image
at shape (1, 3, 224, 640)

0: 224x640 16 persons, 7.7ms
Speed: 1.8ms preprocess, 7.7ms inference, 1.4ms postprocess per image
at shape (1, 3, 224, 640)

0: 384x640 8 persons, 1 bottle, 2 chairs, 8.3ms
Speed: 2.3ms preprocess, 8.3ms inference, 1.4ms postprocess per image
at shape (1, 3, 384, 640)

0: 256x640 11 persons, 7 chairs, 4 laptops, 7.6ms
Speed: 1.9ms preprocess, 7.6ms inference, 1.3ms postprocess per image
at shape (1, 3, 256, 640)

0: 448x640 14 persons, 2 backpacks, 1 cup, 5 chairs, 1 laptop, 10.1ms
Speed: 3.9ms preprocess, 10.1ms inference, 1.5ms postprocess per image
at shape (1, 3, 448, 640)

0: 448x640 15 persons, 3 chairs, 10 laptops, 6.7ms
Speed: 3.4ms preprocess, 6.7ms inference, 1.3ms postprocess per image
at shape (1, 3, 448, 640)

0: 448x640 10 persons, 1 tie, 1 bottle, 1 cup, 2 chairs, 1 tv, 1
laptop, 6.8ms
Speed: 3.5ms preprocess, 6.8ms inference, 1.3ms postprocess per image
at shape (1, 3, 448, 640)

0: 448x640 15 persons, 1 chair, 2 laptops, 6.9ms
Speed: 3.7ms preprocess, 6.9ms inference, 1.3ms postprocess per image
at shape (1, 3, 448, 640)

0: 448x640 10 persons, 1 tie, 1 cup, 9 chairs, 2 dining tables, 5
laptops, 6.9ms
Speed: 3.8ms preprocess, 6.9ms inference, 1.3ms postprocess per image
at shape (1, 3, 448, 640)
```

```
0: 448x640 14 persons, 7 chairs, 1 dining table, 6 laptops, 7.1ms
Speed: 3.5ms preprocess, 7.1ms inference, 1.4ms postprocess per image
at shape (1, 3, 448, 640)

0: 448x640 16 persons, 7 chairs, 6 laptops, 7.1ms
Speed: 3.4ms preprocess, 7.1ms inference, 1.4ms postprocess per image
at shape (1, 3, 448, 640)

0: 448x640 11 persons, 1 backpack, 1 handbag, 1 tie, 1 bottle, 1 cup,
2 chairs, 1 laptop, 7.0ms
Speed: 3.5ms preprocess, 7.0ms inference, 1.3ms postprocess per image
at shape (1, 3, 448, 640)

0: 448x640 18 persons, 1 handbag, 1 bottle, 7 chairs, 6 laptops, 6.7ms
Speed: 3.4ms preprocess, 6.7ms inference, 1.3ms postprocess per image
at shape (1, 3, 448, 640)

0: 480x640 16 persons, 1 handbag, 2 bottles, 7 chairs, 2 dining
tables, 1 laptop, 7.4ms
Speed: 3.5ms preprocess, 7.4ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 3 persons, 1 backpack, 1 handbag, 1 chair, 1 dining table,
1 laptop, 6.8ms
Speed: 3.4ms preprocess, 6.8ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 3 persons, 1 backpack, 1 handbag, 3 chairs, 1 laptop, 7.6ms
Speed: 3.4ms preprocess, 7.6ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 11 persons, 1 cup, 11 chairs, 5 dining tables, 7 laptops, 1
cell phone, 7.1ms
Speed: 3.6ms preprocess, 7.1ms inference, 1.4ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 12 persons, 2 cups, 12 chairs, 4 dining tables, 6 laptops,
1 cell phone, 6.7ms
Speed: 3.7ms preprocess, 6.7ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 640x480 3 persons, 1 cup, 2 chairs, 1 dining table, 3 laptops,
7.8ms
Speed: 3.8ms preprocess, 7.8ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 480x640 8 persons, 1 bottle, 1 cup, 9 chairs, 2 dining tables, 7
laptops, 7.3ms
Speed: 3.8ms preprocess, 7.3ms inference, 1.3ms postprocess per image
```

```
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 1 cup, 2 chairs, 2 dining tables, 1 laptop, 1
cell phone, 1 book, 7.1ms
Speed: 3.8ms preprocess, 7.1ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 640x480 3 persons, 2 chairs, 1 laptop, 7.4ms
Speed: 3.7ms preprocess, 7.4ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 480x640 12 persons, 12 chairs, 2 dining tables, 5 laptops, 7.2ms
Speed: 3.8ms preprocess, 7.2ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 640x480 2 persons, 1 cup, 2 chairs, 2 laptops, 1 mouse, 1 keyboard,
7.3ms
Speed: 3.7ms preprocess, 7.3ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 480x640 9 persons, 1 bottle, 1 cup, 10 chairs, 3 dining tables, 4
laptops, 8.0ms
Speed: 3.7ms preprocess, 8.0ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 640x480 3 persons, 1 cup, 2 chairs, 1 dining table, 3 laptops, 3
cell phones, 7.8ms
Speed: 3.8ms preprocess, 7.8ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 2 persons, 1 cup, 2 chairs, 2 dining tables, 3 laptops, 1
cell phone, 7.0ms
Speed: 3.8ms preprocess, 7.0ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 2 persons, 1 cup, 1 chair, 4 laptops, 1 cell phone, 7.2ms
Speed: 3.9ms preprocess, 7.2ms inference, 1.4ms postprocess per image
at shape (1, 3, 640, 480)

0: 480x640 14 persons, 1 cup, 9 chairs, 2 dining tables, 4 laptops,
7.5ms
Speed: 3.7ms preprocess, 7.5ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 640x480 3 persons, 1 cup, 3 chairs, 1 dining table, 2 laptops, 1
cell phone, 7.8ms
Speed: 3.8ms preprocess, 7.8ms inference, 1.4ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 2 persons, 1 cup, 2 chairs, 1 dining table, 3 laptops, 1
```

```
mouse, 1 keyboard, 7.1ms
Speed: 3.7ms preprocess, 7.1ms inference, 1.5ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 3 persons, 1 skateboard, 3 chairs, 2 laptops, 6.8ms
Speed: 3.7ms preprocess, 6.8ms inference, 1.4ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 3 persons, 2 chairs, 7.0ms
Speed: 3.8ms preprocess, 7.0ms inference, 1.4ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 4 persons, 3 chairs, 1 laptop, 7.2ms
Speed: 3.9ms preprocess, 7.2ms inference, 1.4ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 3 persons, 1 cup, 1 laptop, 8.6ms
Speed: 3.9ms preprocess, 8.6ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 2 persons, 1 cup, 2 chairs, 1 dining table, 4 laptops, 1
mouse, 7.0ms
Speed: 3.8ms preprocess, 7.0ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 3 persons, 6.9ms
Speed: 3.8ms preprocess, 6.9ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 640x480 4 persons, 7.1ms
Speed: 3.6ms preprocess, 7.1ms inference, 1.3ms postprocess per image
at shape (1, 3, 640, 480)

0: 480x640 4 persons, 1 dining table, 4 laptops, 1 cell phone, 7.6ms
Speed: 3.6ms preprocess, 7.6ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 6 persons, 1 cup, 1 chair, 1 dining table, 1 laptop, 1 cell
phone, 7.2ms
Speed: 3.7ms preprocess, 7.2ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 3 persons, 1 dog, 2 chairs, 1 dining table, 1 laptop, 6.7ms
Speed: 3.8ms preprocess, 6.7ms inference, 1.4ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 2 chairs, 1 dining table, 1 laptop, 7.9ms
Speed: 3.8ms preprocess, 7.9ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)
```

```
0: 480x640 4 persons, 1 cup, 2 chairs, 2 dining tables, 1 laptop, 1
cell phone, 6.8ms
Speed: 3.7ms preprocess, 6.8ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 11 persons, 1 bottle, 1 cup, 7 chairs, 6 dining tables, 7
laptops, 1 cell phone, 1 book, 7.5ms
Speed: 4.0ms preprocess, 7.5ms inference, 1.4ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 1 backpack, 1 handbag, 2 chairs, 1 dining table,
1 laptop, 6.9ms
Speed: 3.7ms preprocess, 6.9ms inference, 1.4ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 11 persons, 1 bottle, 1 cup, 8 chairs, 3 dining tables, 8
laptops, 7.2ms
Speed: 4.0ms preprocess, 7.2ms inference, 1.4ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 1 cup, 2 chairs, 3 dining tables, 1 laptop, 1
cell phone, 1 book, 6.9ms
Speed: 3.6ms preprocess, 6.9ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 1 cup, 2 chairs, 2 dining tables, 1 laptop, 2
cell phones, 6.7ms
Speed: 3.7ms preprocess, 6.7ms inference, 1.4ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 1 handbag, 1 chair, 1 dining table, 1 laptop,
7.1ms
Speed: 3.7ms preprocess, 7.1ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 3 persons, 3 chairs, 1 dining table, 1 laptop, 1 cell
phone, 7.5ms
Speed: 3.9ms preprocess, 7.5ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 3 persons, 1 backpack, 2 chairs, 1 laptop, 6.6ms
Speed: 3.6ms preprocess, 6.6ms inference, 1.4ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 3 persons, 1 backpack, 1 chair, 1 dining table, 1 laptop,
6.8ms
Speed: 3.7ms preprocess, 6.8ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 3 persons, 2 chairs, 1 dining table, 1 laptop, 1 keyboard,
```

```
1 cell phone, 10.0ms
Speed: 4.5ms preprocess, 10.0ms inference, 1.8ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 2 laptops, 1 keyboard, 2 cell phones, 7.2ms
Speed: 3.8ms preprocess, 7.2ms inference, 1.4ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 1 bottle, 2 chairs, 2 laptops, 2 cell phones,
7.0ms
Speed: 3.7ms preprocess, 7.0ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 4 persons, 1 couch, 3 laptops, 2 keyboards, 1 cell phone,
6.9ms
Speed: 3.7ms preprocess, 6.9ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)

0: 480x640 7 persons, 1 bottle, 3 chairs, 2 dining tables, 5 laptops,
1 keyboard, 1 cell phone, 6.9ms
Speed: 3.8ms preprocess, 6.9ms inference, 1.3ms postprocess per image
at shape (1, 3, 480, 640)
```

Saving the results

```
attendances = final_data
df = pd.DataFrame(attendances)
df.to_csv('attendances.csv', index=False)
```