

## Report: Detecting adults or kids using CNNs

The objective of this project is, given an image, to detect if the people and determine whether they are adults or kids using convolutional neural networks.



Several CNNs have already been developed for the task of object detection in an image that can be used in this context.

Given a CNN for object detection, there are two possible approaches:

- Training the CNN with a custom data set from scratch
- Fine tuning an already trained CNN with a custom data set for the specific task considered

The second option is far more convenient since training a CNN from scratch takes a very long time.

In this project we will use Tensorflow's object detection API for the sake of practicality. Since we are using this API, we will implement one of the CNNs it includes for this project. That said, other very performant CNNs for people detection exist such as YOLO.

Since it is relatively accurate, easy to fine-tune and fast to detect, we will be using the Faster R-CNN model.

### 1. Creating the data set

The first step of this project is to collect a dataset of images. Using a python script, I automatically downloaded a set of 100 images resulting from a query on Google Images using the key words "family photo beach". I then manually selected 50 images that represented, relatively clearly and unobstructedly, a group of people with approximately the same amount of adults and kids. Using LabelImg, I then labeled the kids and adults in every image of the data set and saved the annotation as an.xml file that I then converted using another python script into tensorflow records.

The data set used in this project thus comprised of:

- 50 images
- 150 labels
- 75 adult labels
- 75 kid labels



## 2. Setting up the environment

With a data set of labelled images in hand, I then set up a development environment in a GPU-enabled Colaboratory Notebook. I set up the Tensorflow object detection API, imported all necessary modules and files.

## 3. Fine-tuning the pretrained Faster R-CNN model

Using the API I downloaded the Faster R-CNN model and configured it with pre-trained parameters also downloaded using the API.

Once the model configured, using an script from the API the model is trained. The training takes 15 to 20mn with the GPU accelerated notebook environment.

Once trained, the model's inference graph is exported.

## 4. Running inferences on test images

Once the CNN is trained and exported, in a new tensorflow session, the inference graph is used for detection on a set of test images.

With this data set of 50 images and 150 labels, the results are already very satisfying. Only very few misclassifications or misses occur (less than 5%).

Results of these inferences are found in the results folder.

