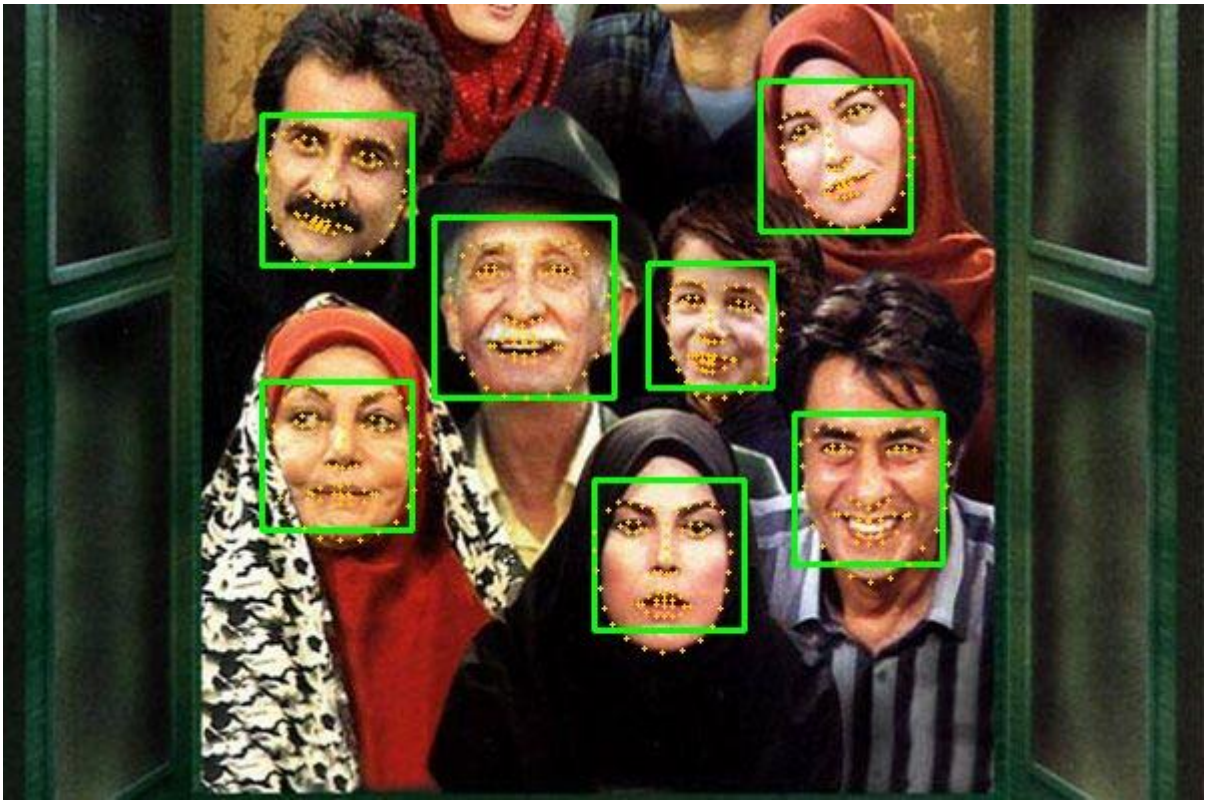
Face Boundary and Facial Landmark Detection Using Convolutional Neural Networks

Alireza Kavian



Overview

Your task is to write a program that detects faces in an image. This detection includes the rectangular bounding box over the faces and the facial landmark key points on them.

Goals

1. Preprocess and prepare training and testing data,
2. Use TensorFlow to create the model architectures,
3. Train a Convolutional Neural Network (CNN) to detect bounding boxes,

4. Train a CNN to detect facial landmarks on the faces

Dataset

You can download the dataset here.

This dataset consists of **4275 images** for training and testing. How to split into train and test data is up to you. For example, you can split it into 75% training data and 25% test data.

In this dataset, you can find *annotations.txt* which contains the labels including bounding box and landmarks annotations. The **format** of this text file is as follows (line by line):

- *x and y* represent the top-left point of the bounding box
- *w* and *h* are the width and height of the bounding box
- *(x1, y1) up* to *(x68, y68)* are the 68 landmark positions in the face

Phase 1: bounding box detection

In this phase, you need to:

1. Read input data and preprocess to create training and testing data,
2. Make a CNN architecture using pure TensorFlow (not even Keras let alone Pytorch),
3. Create a model and start the training process,
4. Test and display the history of the training results.

Notes

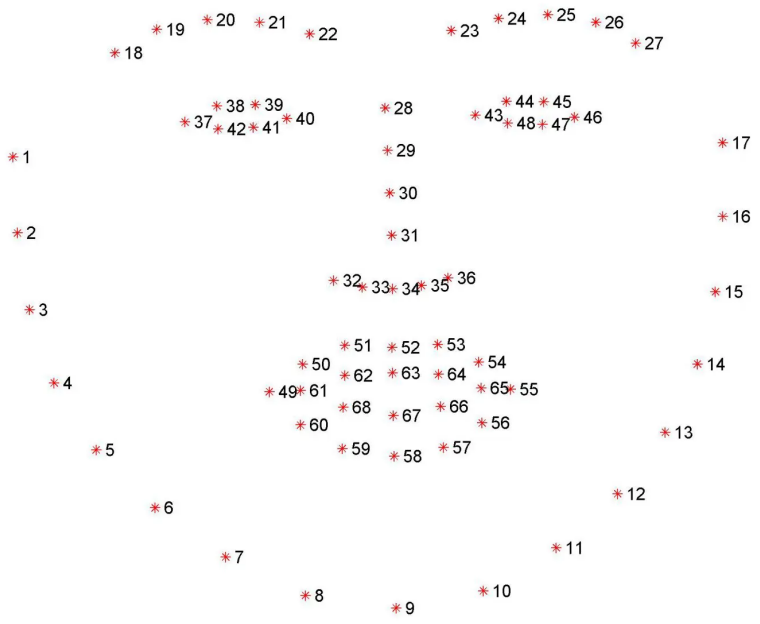
- You need to read only the bounding box annotations for this phase.
- Each image may contain **multiple faces**.

- To detect **multiple bounding boxes** in a single network you can use the *SSD* or *YOLO* algorithms and customize the way it functions.
- The bounding box annotations demonstrate only the top leftmost point and also the width and height of the rectangular box.

Phase 2: Facial landmark detection

Like phase one, train a CNN to detect the 68 facial landmarks of a face.

If you are not about to make a single-architecture network (as extra credit), you might want to **crop** the faces and resize them to a common size (transform the landmark points accordingly).



Phase 3: Combination of the results

You need to combine **the results** of phases 1 and 2: Show faces in an image by displaying the bounding boxes and landmarks; just like the picture shown at the top of this document.

Teamwork and Presentation

- You may work in teams of at most 2 students. However, each student will present the project to the TAs individually. Each student must be able to fully explain all parts of the code.
- Besides the other parameters, your program is evaluated by testing multiple images visually at the time you are presenting it to your TA.
- The program must be able to detect your own face on a webcam stream. So please have a webcam handy at the time of the presentation.
- **Your programs will be checked for similarity. Similar codes will not be marked.**
- **You cannot use pre-trained networks by no means.**

Extra Credit

Extra credit is given for

1. **Real-time** execution
2. Having a combined **single architecture** for both phases
3. A top-notch **accuracy** for your model(s)

Useful Links

https://ruslanmv.com/blog/Neural-Networks-in-Tensorflow

https://towardsdatascience.com/implementing-ssd-in-keras-part-i-network-structure-da3323f11cff

Why using Tensorflow without Keras: