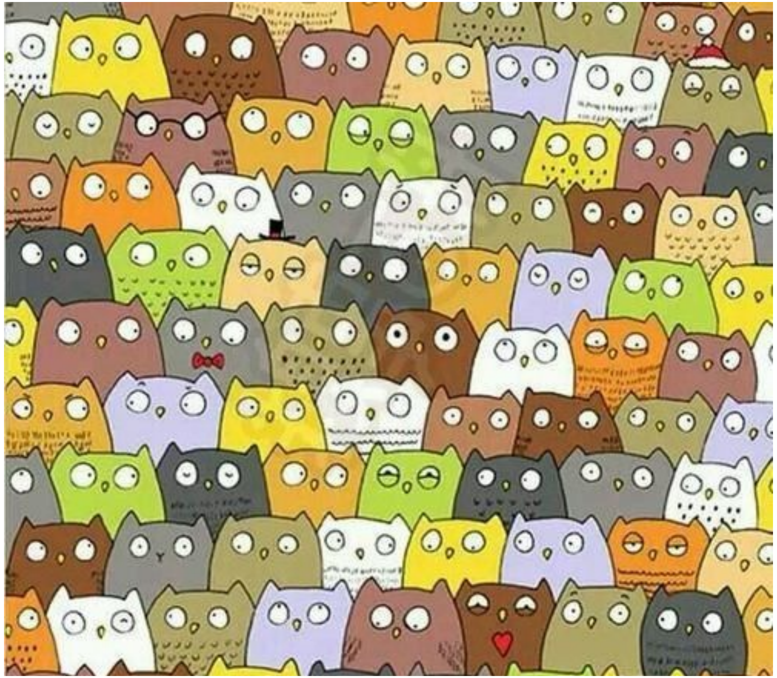
Human face detection

Long Lin, Sina Borgi, Weiyuan Chen, and Malou Maria Nielsen

Outline

- Motivation
- Dataset
- Models
- Results
- Discussion
- Summary

Motivation



Where is the cat?

1. We are all interested in object detection
2. Human face detection is most relevant and practical

Dataset--Kaggle Human Faces

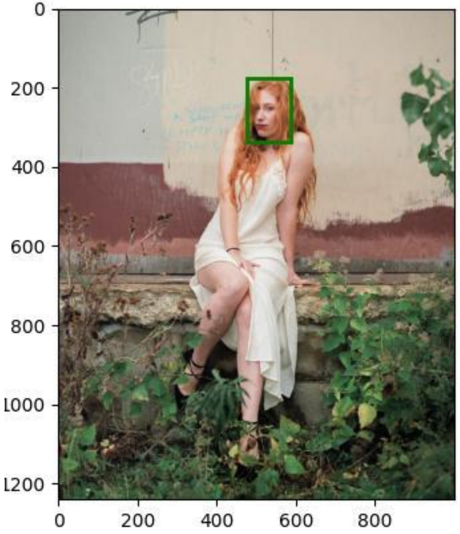
A diverse compilation of human facial images encompassing various **races, age groups**, and **profiles**. (N=2,204)

- High resolution
- Labeled images
- Different sizes
- Different number of faces
- Structured

Dataset--Kaggle Human Faces

A diverse compilation of human facial images encompassing various **races, age groups**, and **profiles**. (N=2,204)

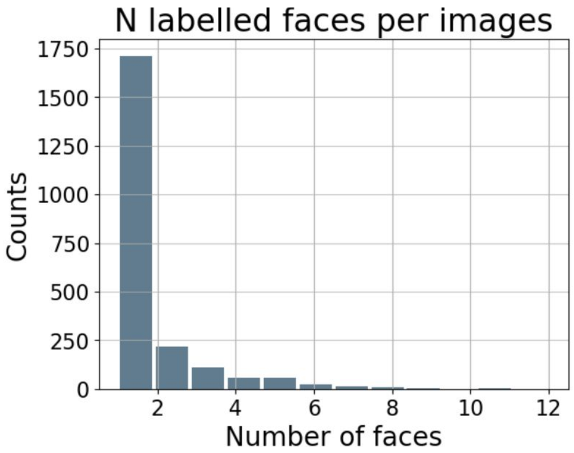
- High resolution
- Labeled images
- Different sizes
- Different number of faces
- Structured

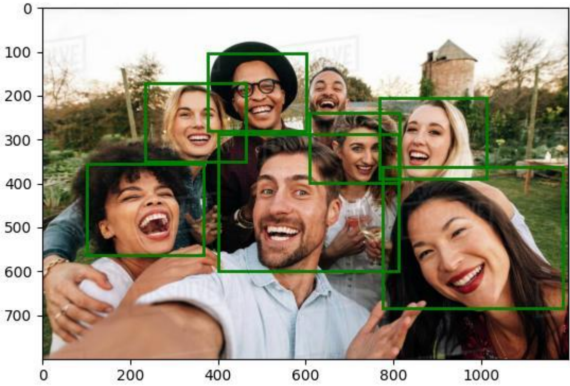


Dataset--Kaggle Human Faces

A diverse compilation of human facial images encompassing various **races, age groups**, and **profiles**. (N=2,204)

- High resolution
- Labeled images
- Different sizes
- Different number of faces
- Structured

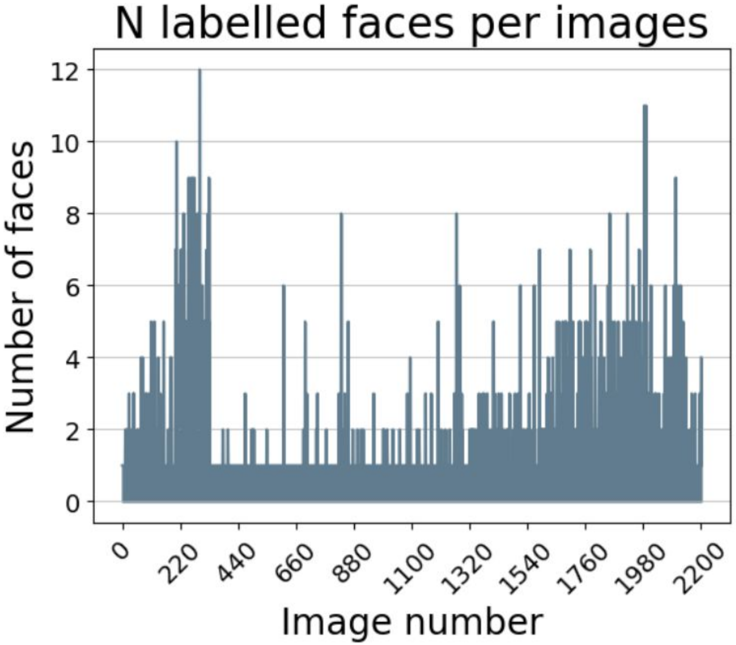




Dataset--Kaggle Human Faces

A diverse compilation of human facial images encompassing various **races, age groups**, and **profiles**. (N=2,204)

- High resolution
- Labeled images
- Different sizes
- Different number of faces
- Structured



Pre-trained Models

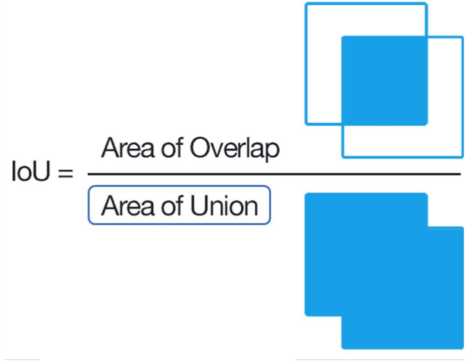
- InceptionResnetV2
  - CNN -- 164 layers
  - Trained on more than a million images (No human faces)
  - Classify images into 1000 object categories
  - Input size of 299-by-299
- Xception
  - CNN -- 71 layers
  - Trained on the same images dataset as above (No human faces)
  - Classify images into 1000 object categories
  - Input size of 299-by-299
- MTCNN
  - Multi-task Cascaded Convolutional Networks
  - Combined 3 CNNs for face classification, bounding box regression, facial landmark localization

Pre-trained models

Model	Accuracy	Average_IoU
InceptionResnetV2	0%	\
Xception	0%	\
MTCNN	93%	0.43

Accuracy: $\dfrac{\frac{\sum true\ positive\ rate\ per\ image}{N}}{N}$

IoU(Intersection over Union):



Implementation

- Pre-trained on Imagenet
    - Classify 1000 objects or animals
- Removed top layers (*bottom)
    - Adding our own
- Output:

how many faces

$4 * max\_number\_faces + 1$

corners of the box

- Accuracy: IoU

Figure 5. Schematic diagram of InceptionResNetV2 model (compressed view).

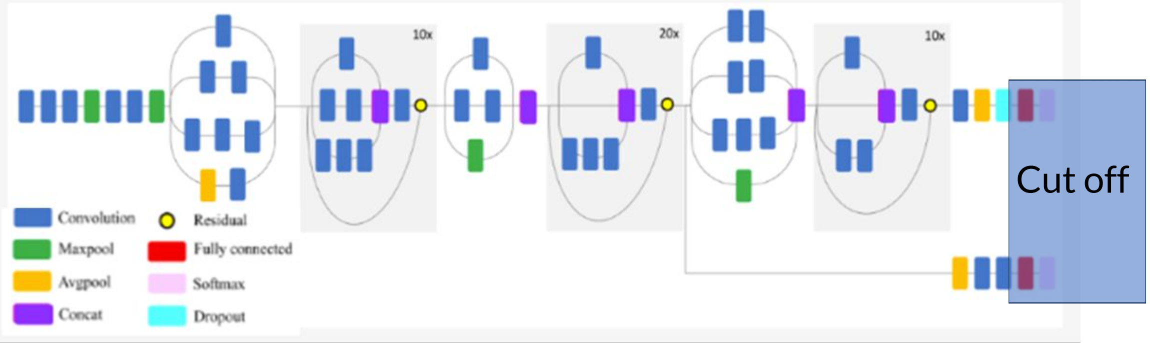
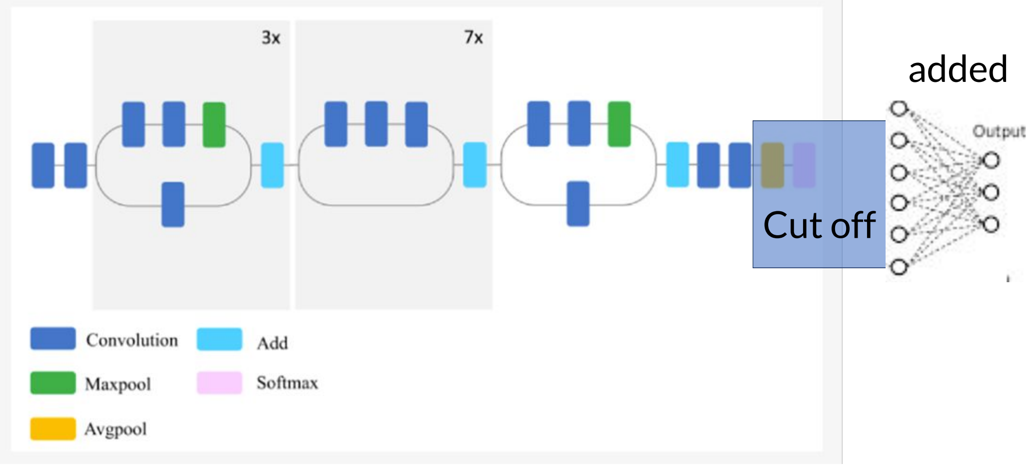
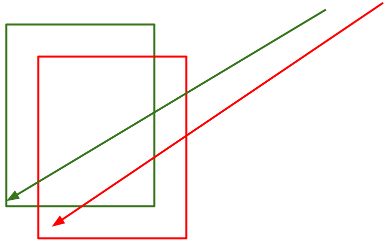


Figure 4. Schematic diagram of Xception model (compressed view).



Implementation

- Loss: MSE = $(1/n) * \Sigma(x_i - x)^2$



- Training process
  - Train the added layers (Frozen)
  - Retrain with all layers (Unfrozen)

Figure 5. Schematic diagram of InceptionResNetV2 model (compressed view).

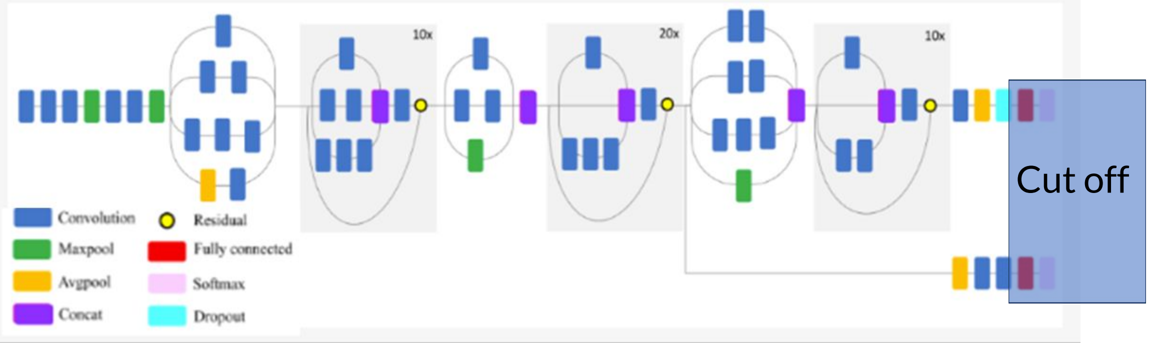
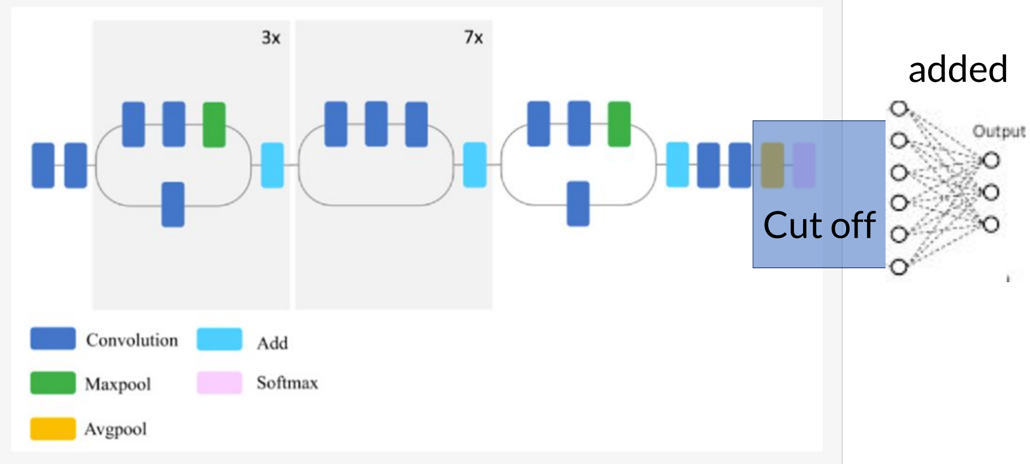
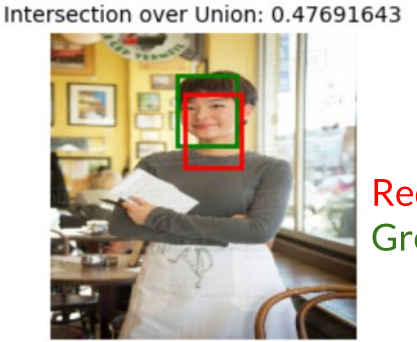


Figure 4. Schematic diagram of Xception model (compressed view).

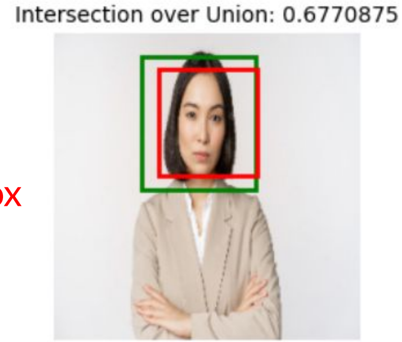


Results: Xception

- Average IoU(Frozen): 0.5
- Average IoU(Unfrozen): 0.7
    - Trained on sharp images



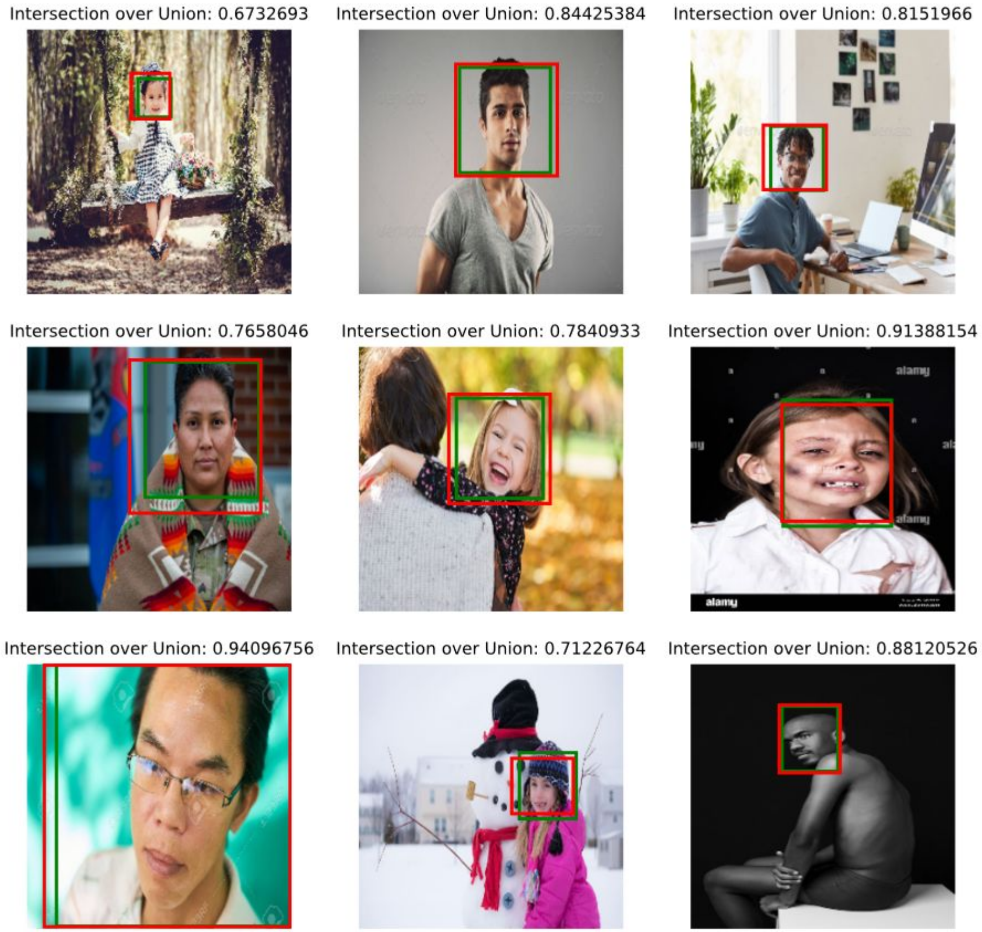
Red: Predicted box
Green: True box



InceptionResnetV2

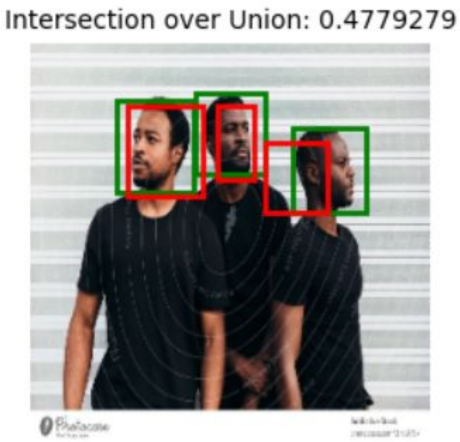
- Average IoU (Unfrozen): 0.792

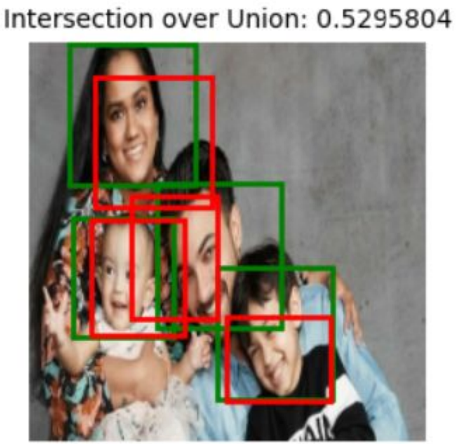
- Decent performance
  - Single faces
  - Trained on clean images

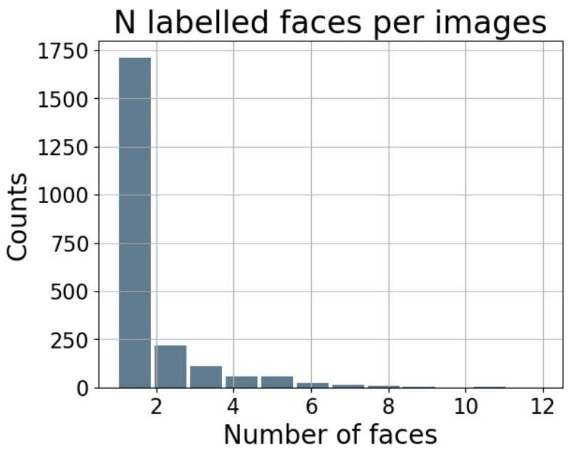


InceptionResnetV2

- Not so decent performance
    - Multiple faces

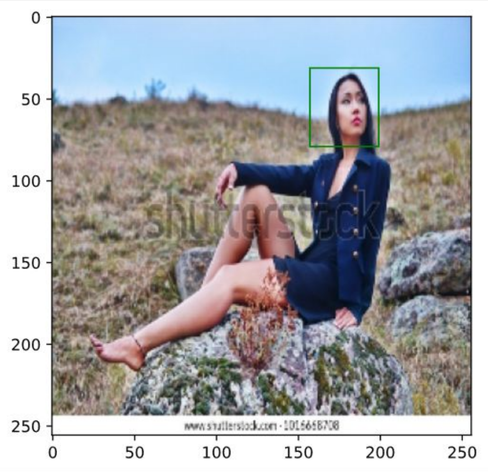






InceptionResnetV2: Are all images perfect?

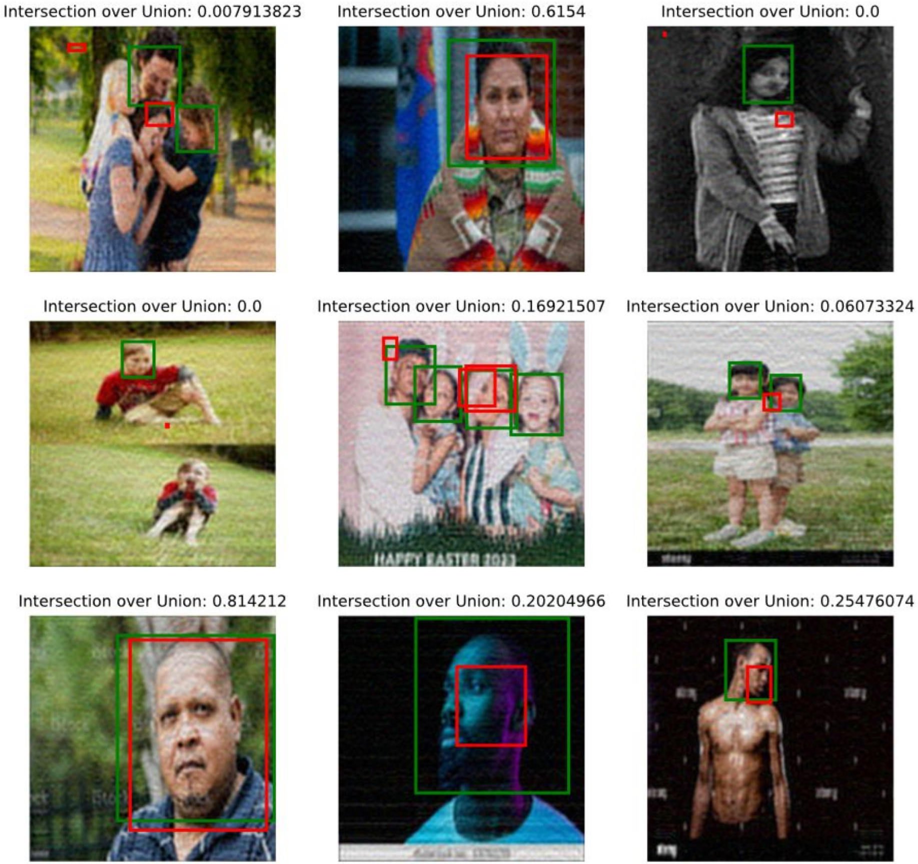
Blur it until humans almost can't distinguish



Can the model still recognize faces?

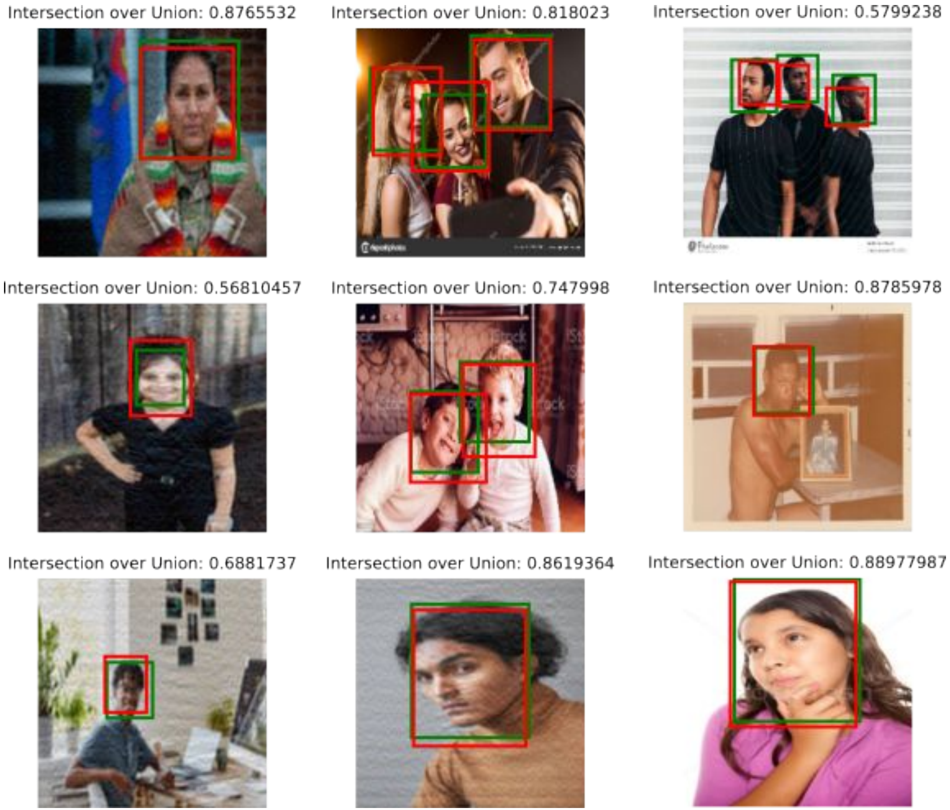
- Sometimes..

- Different training needed



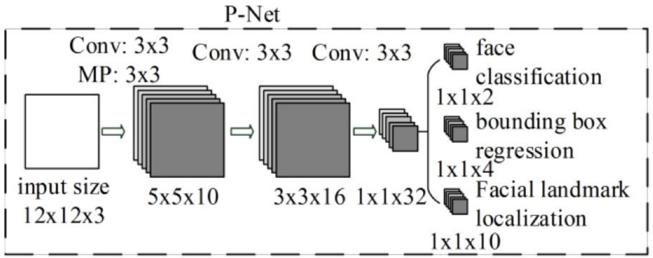
Can the model still recognize faces?

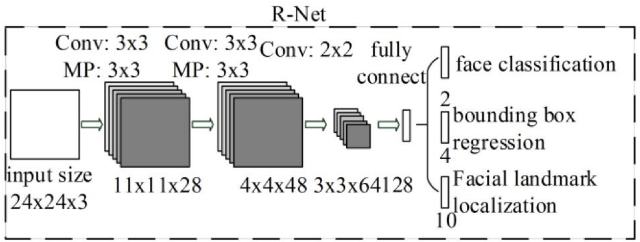
- Sometimes..
- Different training needed
  - Pre-processed + clean images
- Accuracy:
- Average IOU: 0.824 (+ 0.032)

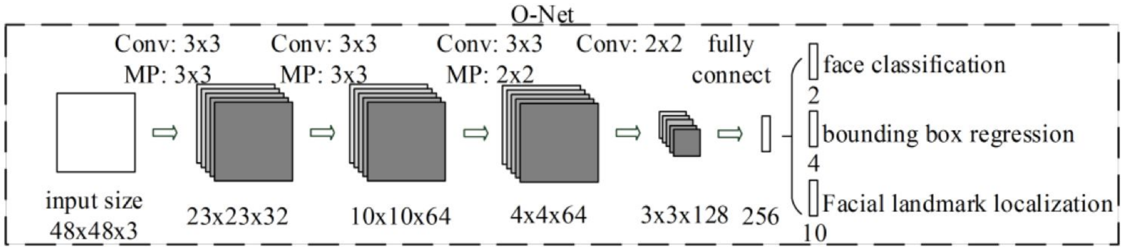


MTCNN – Multi-Task Cascaded Convolutional Neural Network

- The 3 stages:
  1. The proposal network (P-Net)
  2. The refine network (R-Net)
  3. The output network (O-Net)
- The 3 tasks:
  1. Face classification
  2. Bounding box regression
  3. Facial landmark localization
- **Zhang et al. 2016**



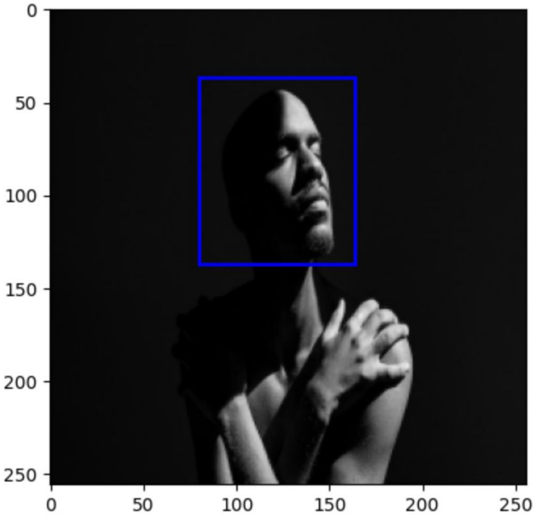


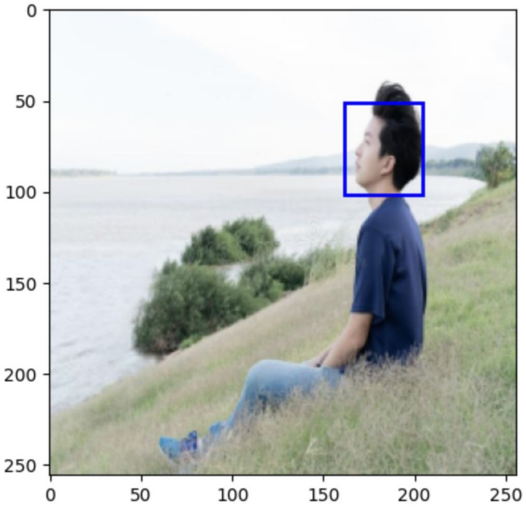


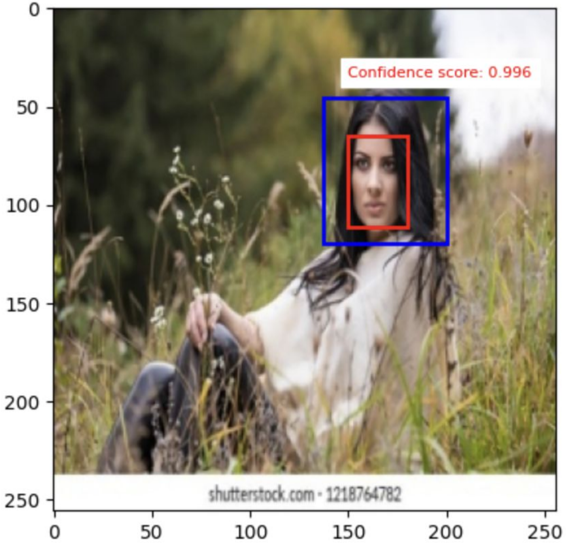
MTCNN – Multi-Task Cascaded Convolutional Neural Network

Problems:

- Sometimes it can't find any faces .
- Difference to the true boxes (low IoU, but is it bad?)



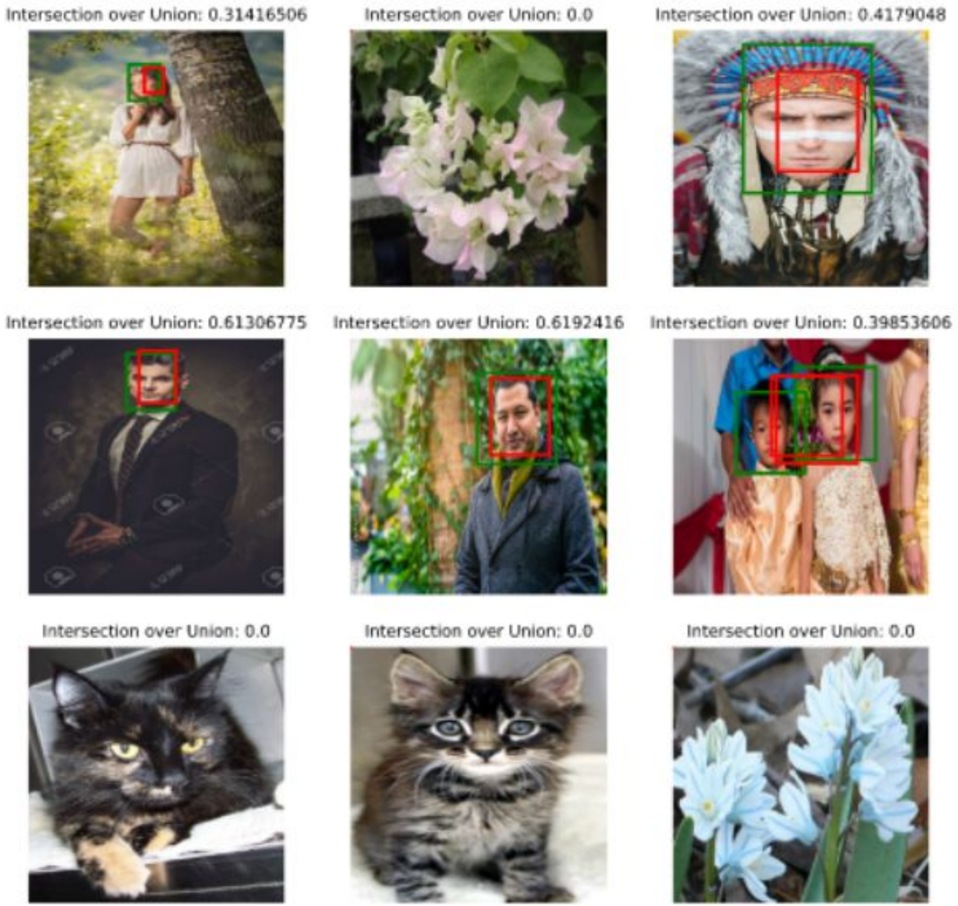




IoU = 0.302
Red: Predicted box
Blue: True box

Discussions

- Good at individuals
- Labels
  - Not the best
- Overtraining
  - Maybe?
- Weaknesses
  - Multiple faces
  - Lower resolution (and small)
- Lacking comparability



Summary

1. InceptionResnetV2(Retrained)
    a. Powerful on individual faces
    b. Weaker on multiple
    c. Stronger when trained on pre-processed images

2. Xception(Retrained)
    a. Decent on individual faces
    b. Weaker on multiple

3. MTCNN
    a. Powerful on multiple faces
    b. Independent of labels

Thank you for listening

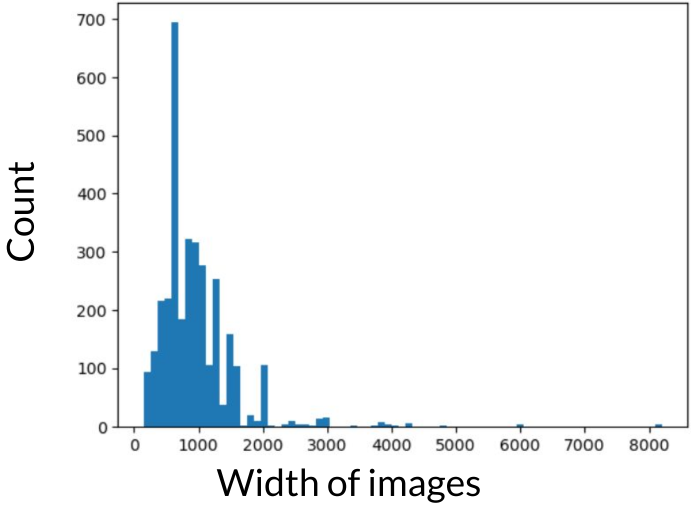
Appendix

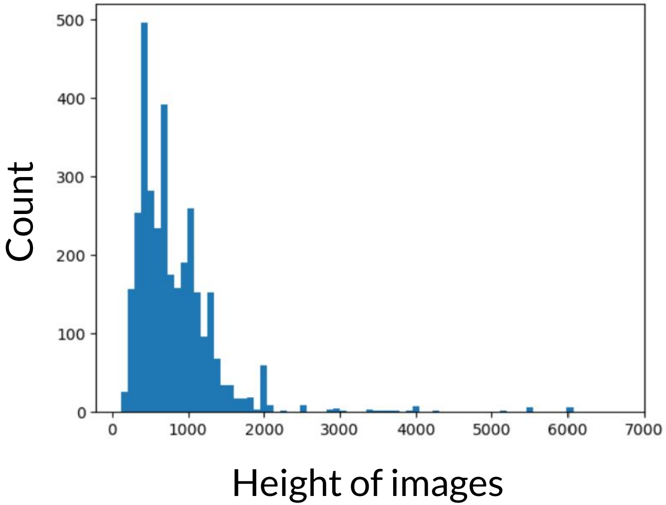
Motivation

In this project we wanted to find a dataset of images. Find an appropriately pre-trained model that could take images and do object detection on it. If possible the final model should be able to draw a box around human faces. Once this step has been done we can see if it is possible to finetune the initial model and get better and better guesses on where the human faces are, or if they are not there.

Data

This project will take a dataset of images from Kaggle (N = 2204). The data is photographs of people (individuals and groups), and the goal of this project is to find a pre-trained model, or multiple, to draw boxes around human faces. The data comes in different sizes as seen in the Figures below.





Data

The data is also labelled with bounding boxes that gives the x_0, y_0, x_1 and y_1, coordinates of the top-left and bottom-right corners of the box around each face in each image (shown in Fig 1). This means that images with multiple faces has multiple labels (shown in Fig 2).

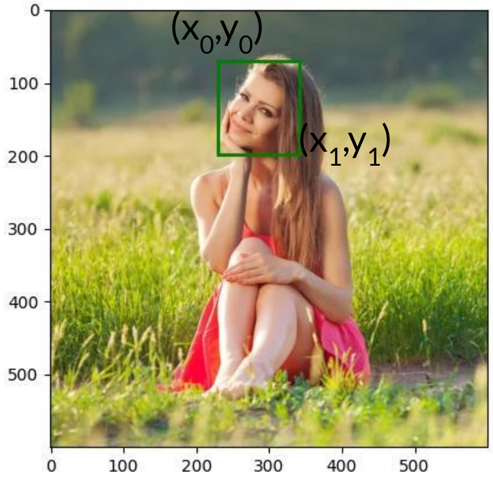


Fig 1

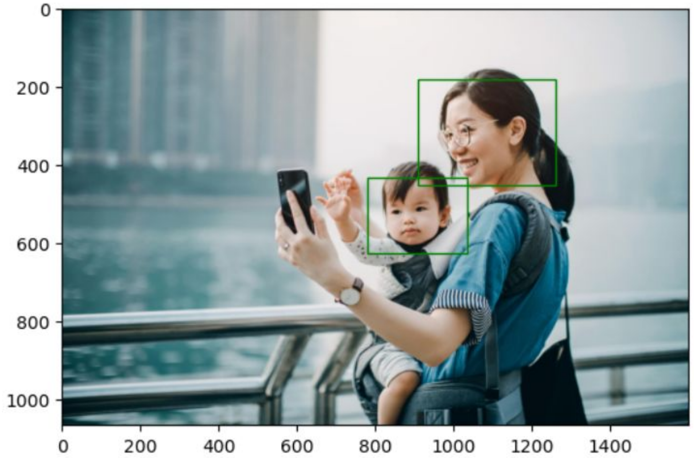
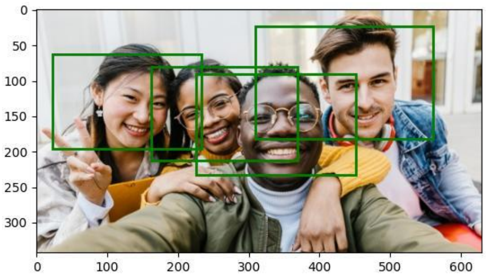


Fig 2

Data

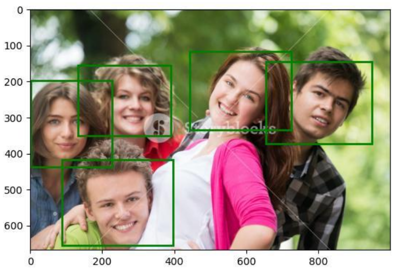
The coordinates were produced by ssd_mobilenet_v2_face_quant_postprocess model. There are some issues that may affect our training:

1. Overlapping between multiple faces coordinates. e.g. 00000562.jpg
2. Some images don't have all labels for all faces, possibly due to the lack of power of that model, e.g. 00000616.jpg
3. Some images have more labels than faces. e.g. 00002857.jpg
4. Duplicated/highly similar images. e.g. 00000280.jpg vs 00000377.jpg



00000562.jpg

00000616.jpg



00000280.jpg

00000377.jpg

Data source (all Kaggle)

- Human faces: https://www.kaggle.com/datasets/sbaghbidi/human-faces-object-detection

- Flowers: https://www.kaggle.com/datasets/prasunroy/natural-images
  - Subfolder:/flower

- Cats: https://www.kaggle.com/datasets/prasunroy/natural-images
  - Subfolder:/cat

Methods and thoughts: InceptionResnetV2

Then by adjusting the pre-processing phase, training of the model and doing hyper parameter tuning. The aim is to improve on basic pre-trained models accuracy on detecting faces. For InceptionResnetV2, hyperparameter tuning included tests on the learning rate, how many layers to add to the end of the pre-trained model and how many nodes on them. The optimal settings were found using a “grid” with lr = 0.0001, 2 extra hidden layers of 256 and 128 nodes. Cross validation was not working with the way that the data/labels were setup. We had to use the:

ds = tf.data.Dataset.from_tensor_slices(images_path).map(lambda x: tf.numpy_function(load_image_and_boxes, [x], [np.float32, np.float32]))

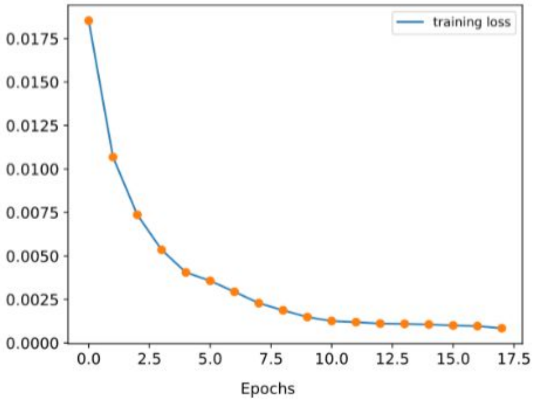
Which caused model.fit(validation_data=val_data) to fail same with validation_split

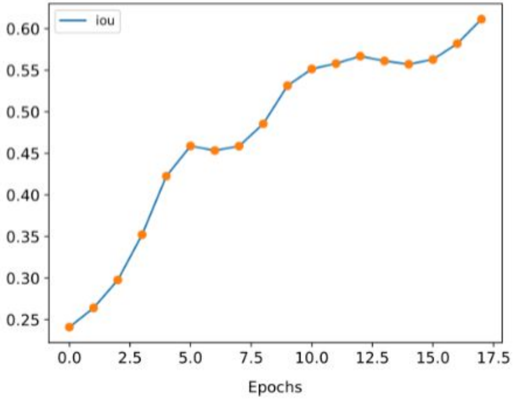
Methods and thoughts: InceptionResnetV2

However, pre-processing was tried as a measure of improving the detection capacity of the model. Here computer vision functions such as blurring matrices, high&low pass Fourier filters were used to mess up the images. By training the model on both clean and pre-processed images the over IoU and loss functions were better. This was could have been because the versatility of different types of images made the model more robust. At least towards less “optimal” images where the there isn’t only 1 face, centered, focused, and looking at the camera and with higher resolution.

Initial training with only the added layers

On the right, the loss and IoU score can be seen for each epoch of the initial training. This is where the end of the pre-trained model has been cut-off and the extra hidden layers has been added with the desired output layer. Clearly the weights are way off in the beginning, but without much training there is a huge improvement after just 10 epochs (on pre-processed images).



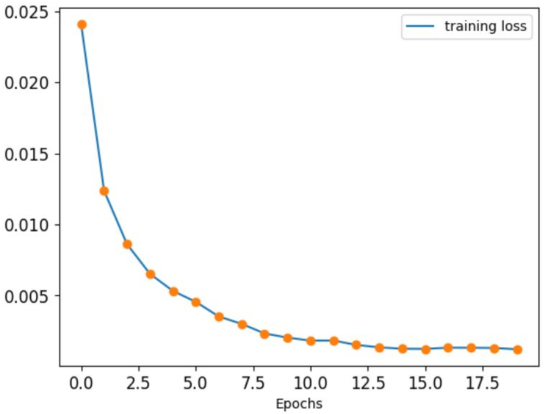


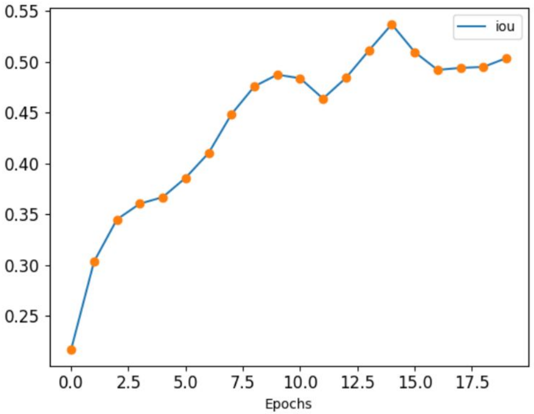
Methods and thoughts: Xception

Xception was trained with the same parameters as InceptionResNetV2.

The figures on the left show loss and IoU scores for the initial training with the added layers.

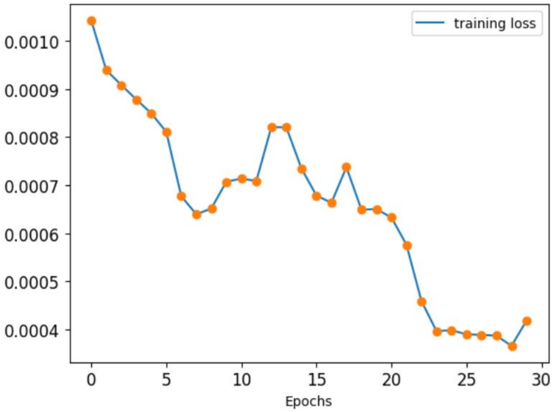
The IoU score improved until it was around 0.5.

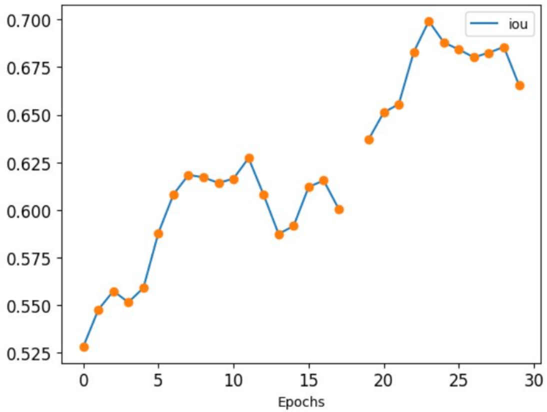




Methods and thoughts: Xception

After training with all the layers, the model saw improvements in IoU score until 23 epochs. At around 17 epochs in the bottom IoU figure there is a gap in the graph. This is due to the output being a NaN, which sometimes occurs because the IoU score is calculated by dividing with the area of union. If the area is 0, then IoU can’t be calculated.





MTCNN - Why did we choose to work with it?

- MTCNN is currently one of the most popular detection models and known to be very accurate.
- As a completely pretrained model it is independent on the true labels, and as discussed the true labels might not be the very best, so an independent model might provide more information.
- We expected it to work really well and the idea was to compare the other models to it.

MTCNN - Preprocessing and implementation

Preprocessing:

- **input size:** doesn’t matter,, but we used 256x256.
- **Color format:** BGR.
- **Integer type:** uint8.

Implementation:

- mtcnn package:

```
model = mtcnn.MTCNN( )
faces  = model.detect_faces(Image)
```

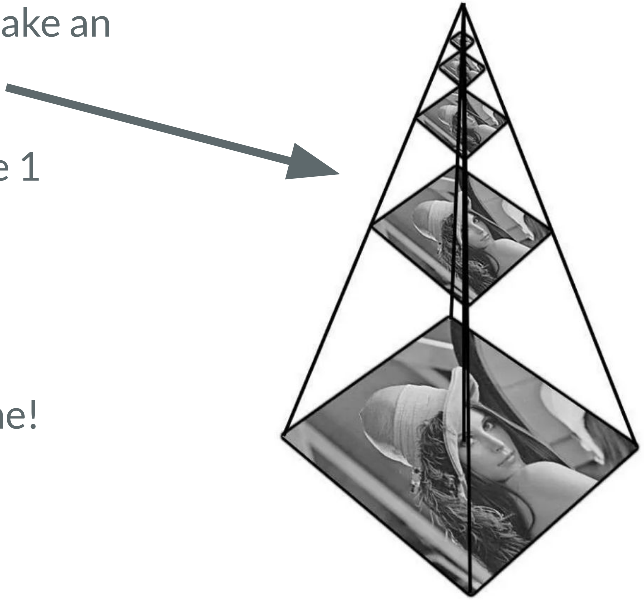
MTCNN - Structure details

The very first step:

- Resizing the image to make an image pyramid.
- This is the input of stage 1 (the P-Net).

The stride of 2:

- Allows for faster runtime!

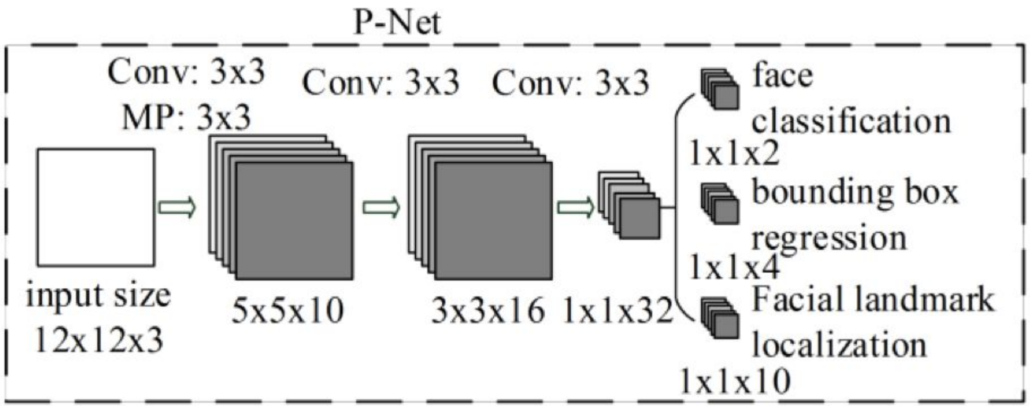


MTCNN - Structure details

Stage 1 (P-Net):

- Fully Convolutional Network (FCN).

1. Finds candidate windows and their bounding box regression vectors.
2. Non-Maximum Suppression (NMS):
   - Highly overlapping candidates are merged.

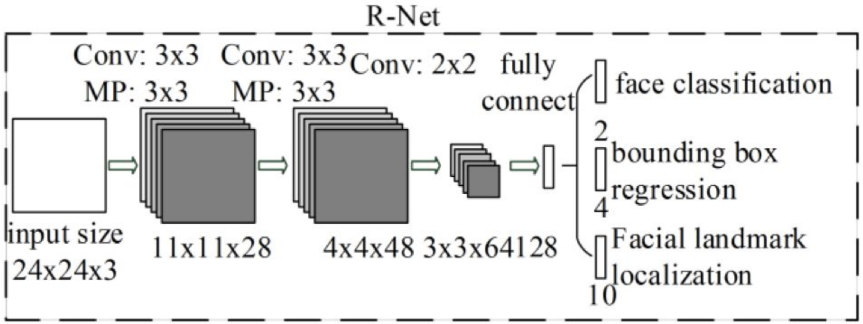


MTCNN - Structure details

Stage 2 (R-Net):

- CNN not FCN

1. It takes the candidates (as 24x24x3 image arrays) from P-Net as input.
2. Low confidence candidates are discarded.
3. Bounding box regression.
4. NMS ones again to discard redundant boxes.



MTCNN - Structure details

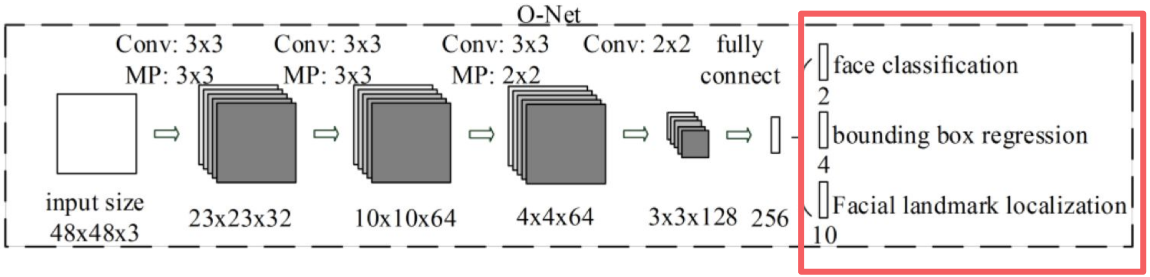
Stage 3 (O-Net):

- CNN not FCN

1. It takes the boxes (as 48x48x3 image arrays) from R-Net as input.
2. Similar to R-Net:
    - Low confidence candidates are discarded.
    - Bounding box regression.
    - NMS.
3. It starts finding facial landmarks.

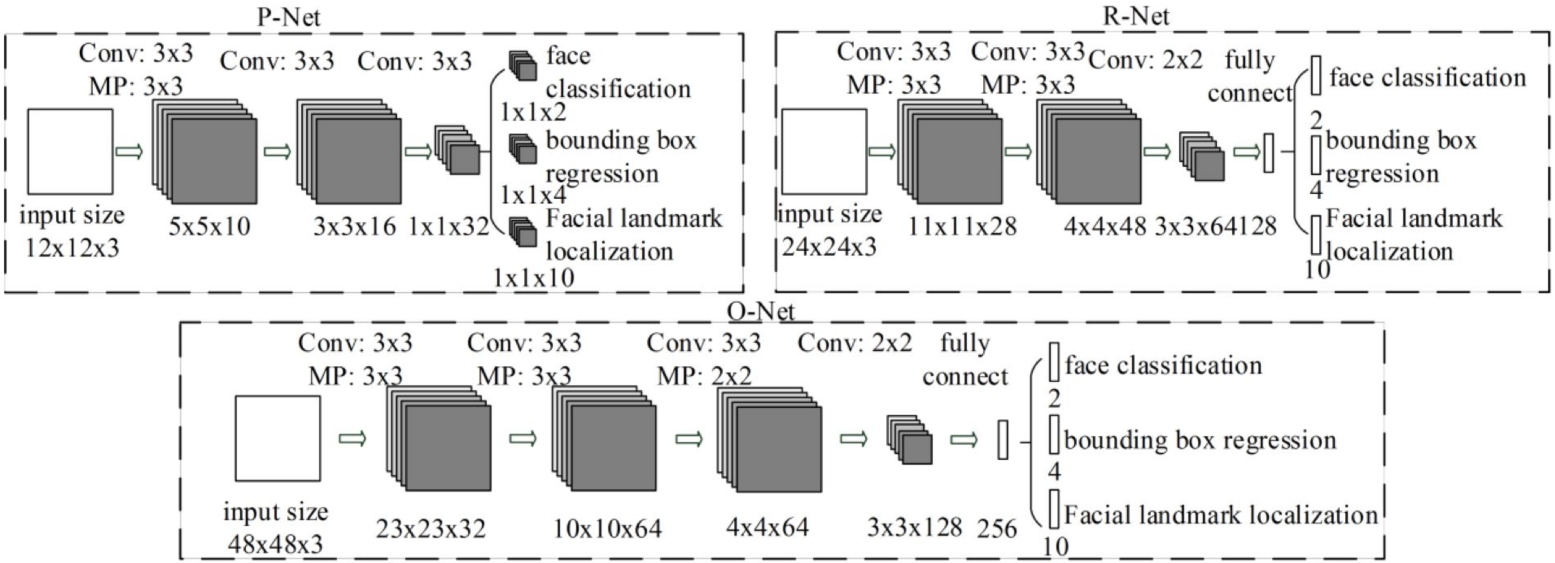
Outputs:

- Face classification (binary classification - is it a face or not?)
- 4 element vector representing the bounding box (x, y, width, height).
- 10 element vector representing 5 facial landmarks.



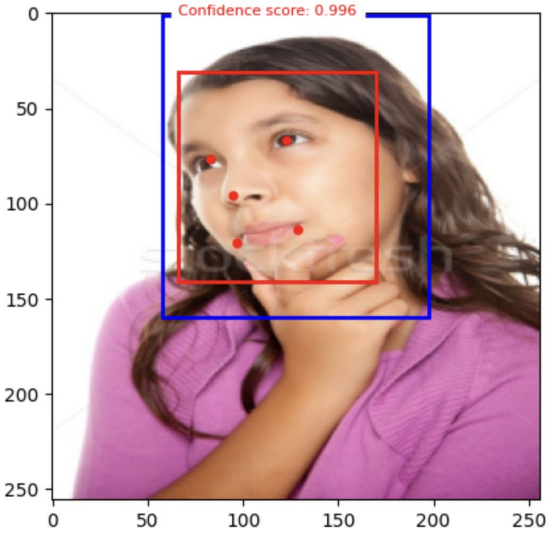
MTCNN - Structure details

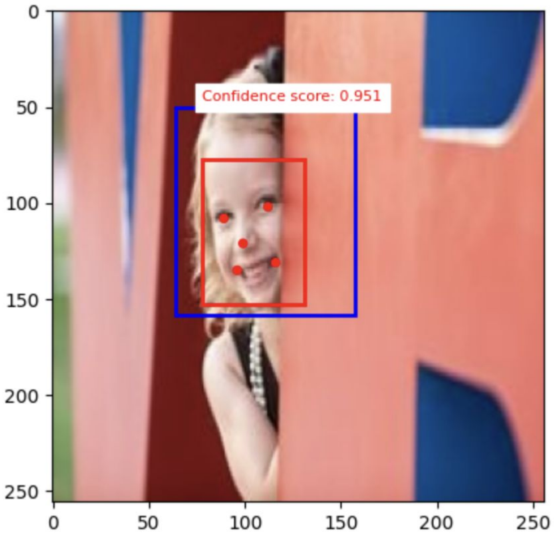
The complete model architecture image taken from Zhang et al. 2016:

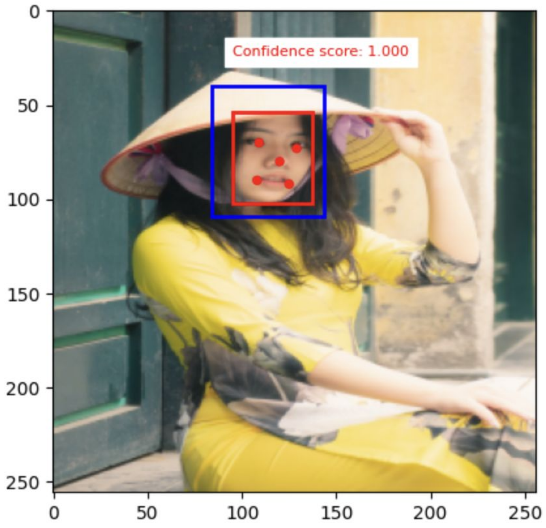


MTCNN - Facial landmarks

- MTCNN also finds 5 facial landmarks (nose, left_eye, right_eye, mouth_left, mouth_right)!
- Since the dataset doesn't have these facial landmarks labeled, we can't quantitatively look at the performance.







MTCNN - How was it trained?

- **Datasets:**
    - **Wider Face**
    - **Celeb A** (has annotated facial landmarks)
    - **Face Detection Dataset and Benchmark (FDDB)**
- **Discussion (Considerations):**
    - Are these diverse in terms of ethnicities, gender, and age?
    - Are these diverse in terms quality (blurring), angles, number of faces in the image, and poses etc.
    - Or is the lack of diversity the reason MTCNN sometimes can’t recognize faces in our dataset?

MTCNN - Sources

- https://arxiv.org/ftp/arxiv/papers/1604/1604.02878.pdf(Zhang et al. 2016)
- https://towardsdatascience.com/how-does-a-face-detection-program-work-using-neural-networks-17896df8e6ff
- https://medium.com/@iselagradilla94/multi-task-cascaded-convolutional-networks-mtcnn-for-face-detection-and-facial-landmark-alignment-7c21e8007923
- https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-ipr.2019.0141

Other models we tried

- Mask R-CNN
  - Amazing results on people and objects but.. deprecated libraries..