

YOLOV4 Age Model instructions.pdf

We have used YOLOV4 pre-trained weights and custom trained them on our dataset for age detection. Our model detects humans and predicts their respective age range. We have custom trained the model against 5 age ranges-

- 0: 0-15
- 1: 16-30
- 2: 31-45
- 3: 46-60
- 4: 60+

Our model would draw a Bounding box over the entire human body and label the age range accordingly. We have split our data into train and test with 90:10 ratio.

Our Dataset -

<https://drive.google.com/file/d/1qih1SdXLV110gnUcpVhuRVJfxY2X9GtL/view?usp=sharing>

Our custom training model has 110 convolutional layers and 3 YOLO layers in total. We experimented by tweaking various parameters. We got best results with-

- batchsize: 64
- Subdivision: 16
- Learning rate: 0.001

The yolov4 model for age detection was custom trained upto 10k iteration, each iteration took a average time of ~6 seconds. For further details of the parameters used, layers, and filters, please refer to the configuration file named 'yolov4-custom' attached above.

Evaluation of Age Model-

We evaluated the model to determine the following metric:

```
IoU threshold = 50 %  
Mean Average Precision (mAP@0.5): 0.998902, or 99.89 %  
F-score:0.99  
Precision: 0.99
```

Implementation-

To train the YOLOV4 model for age detection, follow the below steps-

- Create a folder named 'yolov4_age' in your google drive.
- Create a empty folder named 'training' inside the 'yolov4_age' folder.
- Upload the following to the 'yolov4_age' folder-
 - zip of all the images with their annotation files in YOLO format
 - 'process.py' generating the train/ test split
 - 'obj.names' file containing the labels
 - 'obj.data' file containing the path details of the backup folder, and the train/ test files
 - 'Yolov4-custom' - configuration file containing the details about the model layers, training parameters, batchsize, subdivision etc.
- Upload the 'YOLOV4_Age_Training' file on google colab.
- Connect the notebook to standard GPU and click run all.

Once the training is completed, you can test the model on videos and images. You just need to mention the path to the input video/image. In case of video input, you need to set a path for the video output and in case of image input, the output image will be displayed in the colab itself.