Bosch's Age And Gender Detection

Team 17

Table of Contents

Results of the demo videos provided to us:

https://drive.google.com/drive/folders/1xGAXMLuD8ey6_sZfAOfoky_1-ugCAEAU?usp=sharing

Submission format in the form of CSV file:

https://drive.google.com/drive/folders/1GZD5dCI9yoyYhyJrxsIhGNXju0--Bjir?usp=sharing

Here we have considered frame 80 to frame 130 of the demo videos provided to us for generating the submission CSV file.

Project Overview and Introduction

In the wake of recent advancements in the field of machine learning and artificial intelligence, there is a need for models to accurately classify, summarise, extract and analyse information from massive amounts of data. Consequently, traditional machine learning techniques are not able to cope with the changing trends of data requirements effectively. A crucial step in developing high-end algorithms which have an exemplary generalisation and adaptive power as well resistance to noise is the thorough analysis of

high-dimensional, complex data. To tackle this Deep Learning has proved to be superior to the conventional methods.

The problem statement's task involves estimating the age and gender of a person from surveillance footage. This task not only requires a huge amount of data but also thorough investigation and analysis. Surveillance footage is usually of low resolution which poses a hurdle for models to learn and extract important features. Hence, to tackle this problem a mechanism to convert the low-resolution videos into their corresponding high definition counterparts needs to be developed. Subsequently, a deep learning model capable of predicting the age and gender of multiple people in the said high definition video should be implemented to create a robust pipeline.

The need and relevance of the project are as follows:

- Most existing digital video surveillance systems rely on human observers for detecting specific activities in a real-time video scene. However, there are limitations in the human capacity to monitor simultaneous events in surveillance displays.
- Estimating age and gender from footage can help companies and organisations in keeping a track of their clientele base. It can also serve as backup information with demographic details of the customer in case of theft or robbery.
- Surveillance footage is usually of low resolution which poses a hurdle for models to learn and extract important features.

How to tackle this problem statement?

- A mechanism to convert the low-resolution videos into their corresponding high-definition counterparts needs to be developed.
- Subsequently, a deep learning model capable of predicting the age and gender of multiple people in the said high definition video should be implemented to create a robust pipeline.

Before jumping into the schematics and details of the models we used,

Here is the user manual for final end to end inference.

User manual for Final END_TO_END Inference

Please follow the inference starter guide END_TO_END_inference_final_video.ipynb in the submission folder *MP_BO_T17_CODES* to perform complete end-to-end inference on an input video or image:

"yolov4_age" Age Model Folder link - https://drive.google.com/drive/folders/1gPdRqUNFdJ6CstIByH_Dtcm-oAQGBP-4?usp=sharing

"yolov4" Gender Model Folder link - https://drive.google.com/drive/folders/1uVlIRWDI12TKFdsIjUaMoEJHRo-62Ege?usp=sharing

- Open these two links and then click 'Add Shortcut to My Drive'.
- Then open the "END_TO_END_inference_final_video.ipynb" in Google collab for video input and "END_TO_END_inference_final_img.ipynb" for images inference
- Connect to Standard GPU.
- Rull all the cells one by one following the instructions given in each section.

FAST Srgan generates a high-resolution video output of the input video. YOLO age and gender models then predict the age and gender of multiple people in the high-resolution video. If you wish to check an intermediate video, check the output directory mentioned after each model.

STEP 1: Super Resolution:

What is Super Resolution?

A super-resolution algorithm is trained to generate a Super-Resolution (SR) image for a corresponding Low Resolution (LR) image obtained by adding noise and downsampling a High Resolution (HR) ground truth image. The drawback of most super-resolution techniques is the low perceptual quality of the output image. To overcome this, we have employed SRGAN based algorithms.

SRGAN

SRGAN based algorithms can upsample an image with an upsampling factor of x4. SRGAN is trained to minimise a unique perceptual loss function - a combination of the content loss and an adversarial loss. We have fine-tuned the SRGAN pre-trained model according to our use case using only the validation subset (1087 HR and corresponding 1087 LR images) of the WIDER face dataset. We trained the model for 26,350 epochs until the perceptual loss was reduced to 0.0764. After fine-tuning, the model did preserve the facial features to some extent.

Please refer to the SRGAN folder in the submission folder *MP_BO_T17_CODES* for detailed instructions on fine-tuning and inference. Run the finetuned-SRGAN_vid_inference.ipynb notebook in the SRGAN folder For inference of our trained model.

FAST-SRGAN

This approach follows the existing SRGAN architecture but replaces the residual blocks with inverted residual blocks for faster operation and higher efficiency. We have used the official pre-trained weights of FAST SRGAN for our use case.

Please refer to the FAST SRGAN folder in the submission folder *MP_BO_T17_CODES* and run Fast_SRGAN_img_inference.ipynb and Fast_SRGAN_vid_inference.ipynb notebook in this folder for getting the inference of FAST-SRGAN on an input video. All the instructions are provided in the notebook itself.

ESRGAN

We have used the official pre-trained weights of FAST-SRGAN for our use case.

Please refer to the ESRSRGAN folder in the submission folder *MP_BO_T17_CODES* and run inference.ipynb notebook in this folder for getting the inference of FAST-SRGAN on an input video. All the instructions are provided in the notebook itself.

Comparison

SRGAN techniques	PSNR	MSE	SSIM	DISTS
Baseline fine-tuned SRGAN	36.6	35	0.78	0.01
Fast-SRGAN	40	33.31	0.96	0.0075
Real-ESRGAN	39.83	35.7	0.89	0.0070

Following is the comparison wrt to the upsampling time for some videos of a specific dimension, fps, and duration:

Input video frame dimensions (widthxheight)	Fast-SRGAN frames upsampled per minute	Real-ESRGAN frames upsampled per minute
1280x720	1	<1
320x240	26	2

How are these evaluation metrics calculated?
PSNR is calculated for the LR image and the generated, upsampled SR image.
Metrics DISTS, SSIM, and MSE are for comparing the original, ground truth HR image and the corresponding generated SR image. The lower the SSIM and DISTS, the similar the SR and HR images. The metrics displayed in the table are averaged over multiple images of different dimensions.
DISTS is sensitive to structural distortions but at the same time robust to texture resampling and modest geometric transformations. The fact that it is robust to texture variance is also helpful when evaluating images generated by GANs.

Future Scope:

We could have employed Spatio-temporal approaches to upsample the video. We did try the spatio-temporal Multi-Image SR iSeeBetter model on a sample CCTV footage but it ended up degrading the image quality. We also tried training the iSeeBetter. But it required huge amounts of training data (at least 3,000 seven-frame videos) and large computational power (it takes 13 hrs to train with google collab pro GPU) to observe any significant changes.
One more challenge we faced while testing Fast-SRGAN, was that its 'upsampling time' significantly reduces when the input image size is beyond 384x384.
One possible solution to this could have been to apply the FastSRGAN like a convolutional filter, that would transverse the entire image in parts and produce upsampled images. Then we could combine these images to produce the output for one single image.

STEP 2: Age and Gender Detection:

Facial based CNN model

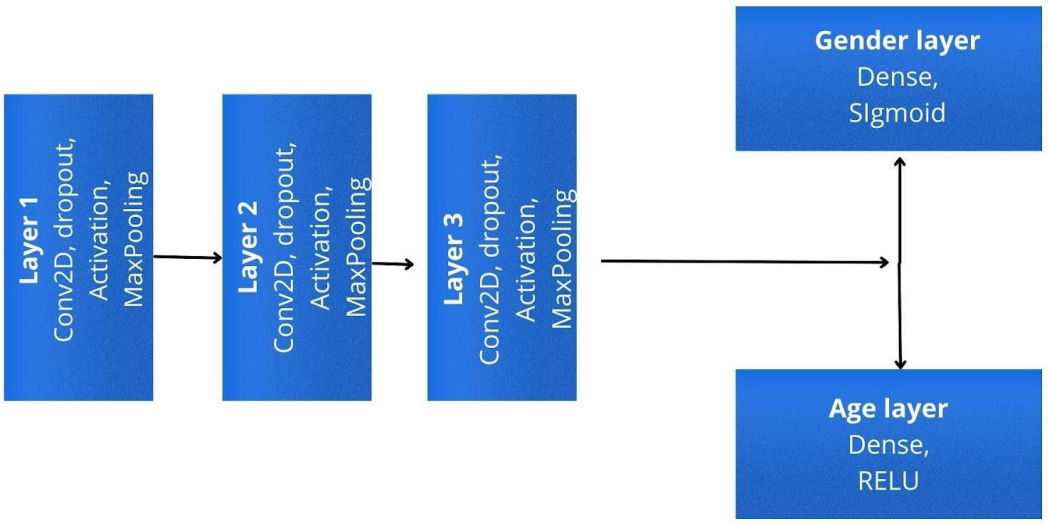
The benefit of using CNNs is their ability to develop an internal representation of a two-dimensional image. This allows the model to learn the position and scale-invariant structures in the data which is important when working with images. We have used a deep convolutional neural network to train our model from scratch on the UTK Face dataset.

Dataset:

- We have used the UTKFace dataset from Kaggle. https://www.kaggle.com/datasets/jangedoo/utkface-new
- UTKFace dataset is a large-scale face dataset with an age spanning from 0 to 116 years. It encompasses over 10000+ cropped face images with annotations of age and gender.

Model:
We used a deep convolutional neural network to train our model on the UTK Face dataset.
Since gender prediction is a binary classification problem and age prediction is a regression problem, sigmoid is used as the output layer for the gender

model and RELU is used as the output activation layer for the age model. Moreover, we have used ‘binary cross-entropy’ as the loss function for gender and ‘mean absolute error’ as the loss function for age.
Here is the schematic of our Deep CNN model-



Evaluation:
Age:
The final trained model traced a linear regression line that roughly passed through the center of the sample distribution when evaluated.
Gender:
Here are the precision, recall, F1 score and accuracy for the gender detection model:

0 (male)	0.93	0.83	0.88
1 (female)	0.84	0.94	0.88

Conclusion:
Even though highly accurate on facial images, these models work poorly on CCTV footages and images. Therefore, to tackle this problem, we propose training the model on full-body data instead of facial images. Additionally, it wasn’t performing well in real-time and required high processing time for videos. Nevertheless, this model serves as a good baseline for the upcoming models. Hence we decided to come up with the YOLOV4 approach which is considered faster.

Please refer to the Facial_age_gender folder in the submission folder *MP_BO_T17_CODES* and run the utkface.ipynb file for training and inference of the model. All instructions are provided in the said notebook. The pre-trained model is also added to the folder.

FINAL PROPOSED APPROACH: YOLOV4 with DARKNET BACKBONE

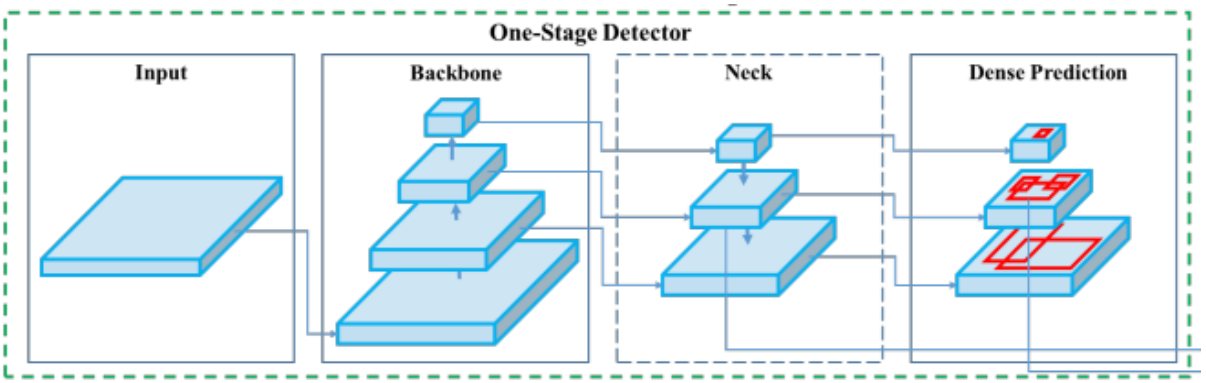
You only look once (YOLO) is a state-of-the-art, real-time object detection system.
YOLO was written in a custom framework called Darknet. Darknet is a very flexible research framework written in low-level languages and has produced a series of the best real-time object detectors in computer vision: YOLO, YOLOv2, YOLOv3, and now, YOLOv4.

Prior detection systems repurpose classifiers or localizers to perform detection. They apply the model to an image at multiple locations and scales. High scoring regions of the image are considered detections. We apply a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

Our model has several advantages over classifier-based systems. It looks at the whole image at test time so its predictions are informed by the global context in the image. It also makes predictions with a single network evaluation unlike systems like R-CNN which require thousands for a single image.

YOLOV4 is a One-stage Detector and its Architecture consists of

- Backbone: CSPDarknet53
- Neck: SPP, PAN
- Head (Dense Prediction): YOLOv3



YOLOV4 One- Stage Detector

Backbone

It refers to the feature-extraction architecture. It is used to improve accuracy, by designing a deeper network to extend the receptive field and to increase model complexity.

Neck

This block is to add extra layers between the backbone and the head (dense prediction block).
In order to enrich the information that feeds into the head, neighboring feature maps from the bottom-up & top-down streams are added together element-wise / or concatenated before feeding into the head. Therefore, the head's input will contain spatially rich information from the bottom-up stream and the semantic rich information from the top-down stream. This part of the system is called the neck.

Head (Dense Prediction)

The network detects the bounding box coordinates (x,y,w,h) as well as the confidence score for a class. The goal of YOLOv4 is to divide the image into a grid of multiple cells and then for each cell to predict the probability of having an object using anchor boxes. The output is a vector with bounding box coordinates and probability classes.

Sub Approach 1: PETA Dataset
Dataset link:
http://mmlab.ie.cuhk.edu.hk/projects/PETA.html

The PETA dataset consists of 19000 images, with resolution ranging from 17-by-39 to 169-by-365 pixels. Those 19000 images include 8705 persons, each annotated with 61 binary and 4 multi-class attributes. Therefore, It can help in recognizing pedestrian attributes, such as gender and age when taking into account the whole body and not the face.

Conclusion: The proposed model was able to learn age and gender information from full-body images but wasn't concerned with facial features. Hence, The accuracy and MA weren't satisfactory as the model could be improved if it was implemented considering facial as well as full-body features.

FINAL Approach: Custom Dataset

Dataset:

We decided to create our own custom dataset for maximum efficiency and performance. Our custom Dataset consists of sample images of pedestrians from all possible angles and their corresponding annotations in the form of bounding box coordinates and labels. We collected over 1300 images of pedestrians in CCTV footage, on busy streets, shops, etc, and annotated them for gender detection. Similarly, we annotated over 500 images for age prediction. The annotations were done with the help of CVAT software.
https://cvat.org/
Our custom dataset is provided:
Age Dataset Link:
https://drive.google.com/file/d/10BC4Tq4Su5RO-T3Zv8BHqe-t8ZwT_Ajk/view?usp=sharing
Gender dataset link:
https://drive.google.com/file/d/1Kbb12koXnUczolr6h9SY9O-xU7GDqHb3/view?usp=sharing

We considered 17 videos for our custom dataset.
Some of the examples of the videos are as follows:

Labels for the dataset are as follows:

For Gender:
0: male
1: female

Age ranges and labels:
0: 0-15
1: 16-30
2: 31-45
3: 46-60
4: 60+

We have used YOLOV4 pre-trained weights and trained them further on our custom dataset. Our custom training model has 110 convolutional layers and 3 YOLO layers in total. We custom-trained various models by tweaking some parameters. We got the best results with-
batch size: 64
Subdivision: 16
Learning rate: 0.001
For further details of the parameters used, layers, and filters, please refer to the configuration file named 'yolov4-custom'.

Evaluation of Age Model:
The yolov4 model for age detection was custom trained up to 10k iterations, each iteration took an average time of ~6 seconds. 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

Evaluation of Gender Model:
The yolov4 model for gender detection was custom trained up to 4k iteration, each iteration took an average time of ~9 seconds. We evaluated the model to determine the following metrics:
MAP: 66.69%
F score: 0.66
Precision:0.66

Inference on CCTV footages:
The proposed YOLO model works well on CCTV footage and can accurately predict the required parameter of a person in real-time. As this model is trained on our custom dataset, we were able to create a model trained especially for the use case of the project.

Conclusion:
The aforementioned YOLOV4 model works the best for the use case of our project as it performs age-gender detection by combining the precise nature of the face recognition strategy with the generalisation power of body-based detection approaches. However, Strict runtime and limited resource constraints on Google Colab did pose a hurdle. Although limited time constraints prevented us from developing a larger custom dataset, we tried our best to accommodate all possible working use cases in our dataset.

Future Scope:
Currently after we process the video through the gender and age model, we are getting 2 bounding boxes- one for Gender and the other for the age range. The age model has to process everything again. Instead, we could only process the parts which are bound by the gender model. This would help enhance its overall performance.

For age- please refer to the yolov4_age folder in the submission folder *MP_BO_T17_CODES* and open the instructions.pdf file for complete instructions regarding the training notebook and dataset downloading procedure.

For gender- please refer to the yolov4_gender folder in the submission folder *MP_BO_T17_CODES* and open the instructions.pdf file for complete instructions regarding the training notebook and dataset downloading procedure.