Team 11

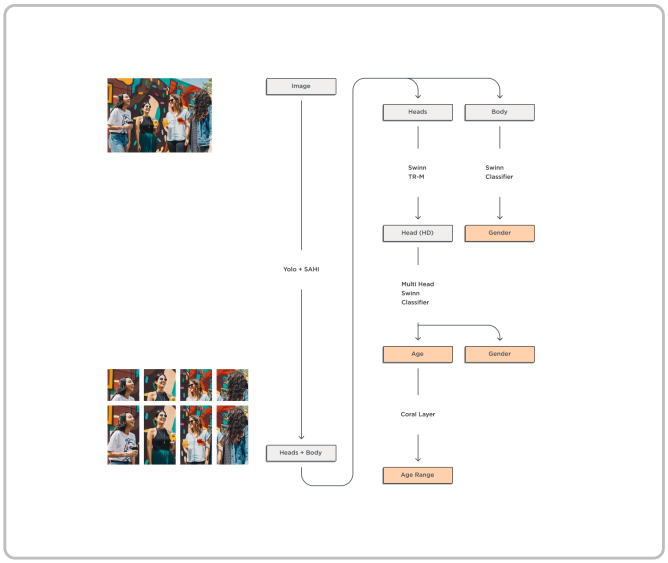
BOSCH'S AGE AND GENDER DETECTION

User Manual

Index

Pipeline Solution

Image Pipeline



Video Pipeline

YOLOv5 + ByteTrack

On taking the video input we first extract individual frames and sequentially pass the frames to an Object Detection model. We have used YOLOv5 finetuned on the Crowd Human dataset. To track the object between frames we use ByteTrack a SOTA tracking algorithm (best MOT results).

Now that we have the bounding boxes we only have the faces for upscaling. This saves computational resources and improves fps.

Models (SWIN with multiple heads)

We then pass the faces to a UTK Face trained SWIN backbone which uses two classifier heads (one for gender and one for age). Using the same backbone improves the efficiency as the model doesn't have to pass through two separate models. Simultaneously, we pass the entire body to a separate model trained on inferring gender from the body. This is helpful as the body gives more useful that only the face doesn't contain.

Averaging over all instances

Finally, we average the age and gender inferred over all the frames in which that particular person is present (using the ids). This way as more of the subject is seen over time we get more accurate results for its age and gender.

Dynamically sized ranges (CORAL Layer)

The PS also requires us to predict a range for age. Instead of having a constant-sized bin for each predicted age, we use the probability scores from the CORAL layer to make a bin we are most confident in. Thus, our bins are dynamically sized based on our confidence in the predicted age.

Setting Up

```
conda create -f environment.yml
conda activate ByteTrack
# On Linux
pip install cython
pip install cython-bbox

# On Windows(not-recommended)

pip install cython
pip install -e git+https://github.com/samson-wang/cython_bbox.git#egg=cython-bbox
# if you face an error in lap pip install
conda install -c conda-forge lap
```

Inference Script

```
python pipeline.py <video_path> #or
python pipeline.py <image_path>
```

csv outputs are saved with the same name as image/video name in ./outputs

Required Arguments

- <video_path>/<image_path>
  - Supported Formats
    - For video : .avi .mp4 .webm .mkv .mov
    - For image : .jpeg .jpg .png .gif

Options

`--save` : Save the annotated video/image in output directory, i.e. outputs

`--display` : View the annotated video/image in real-time

`--hrvid` :Apply upscaling on detected faces (only for video input)

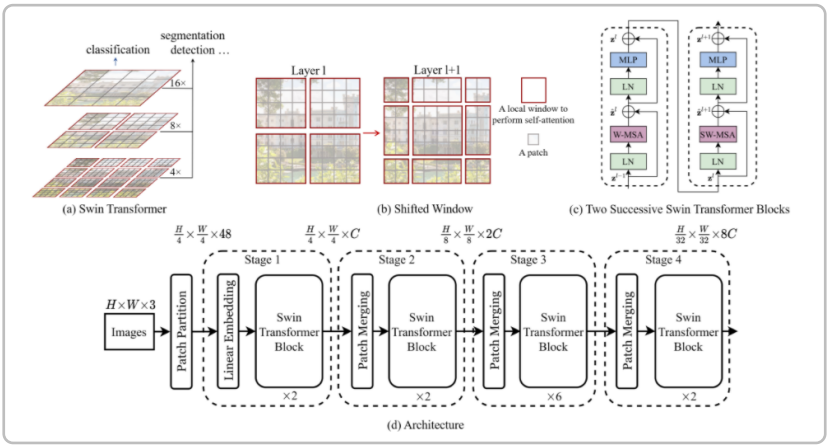
Example Usages

```
python pipeline.py demo_images/sample.jpg
        demo_images/sample.jpg --display      # displays annotated image in opencv window
        demo_vids/video.mp4 --save            # saves annotated video in ./outputs
        demo_vids/video.mp4 --display --hrvid # applies upscaling and displays
```

Models

Swinn

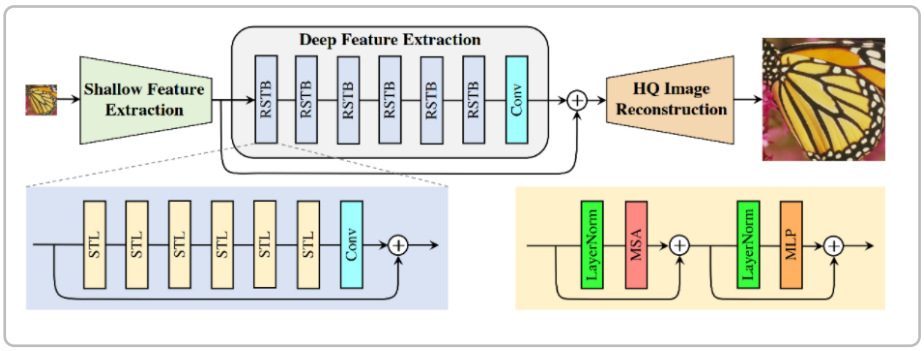
Swin Transformer (the name Swin stands for Shifted window) which capably serves as a general-purpose backbone for computer vision.
Using this model we have trained two classifiers one for the age and other for gender. For the training purpose, both PETA And UTKFace Dataset have been used.



Swinn IR

Swin IR is a strong baseline model for image restoration.

Using this model we have converted the low-quality images from PETA and UTK datasets into high quality. These high-quality images are then trained on Swinn model to predict the age and gender of the person.



Byte Tracker

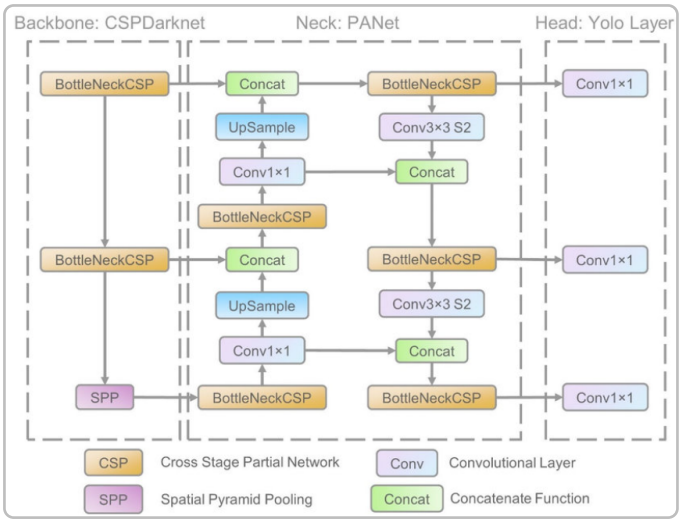
ByteTrack is a simple, fast, and strong multi-object tracker. Multi-object tracking (MOT) aims at estimating bounding boxes and identities of objects in videos.

We have used ByteTrack to get the ids of all the objects in the frame and maintain them between frames.

YOLOv5

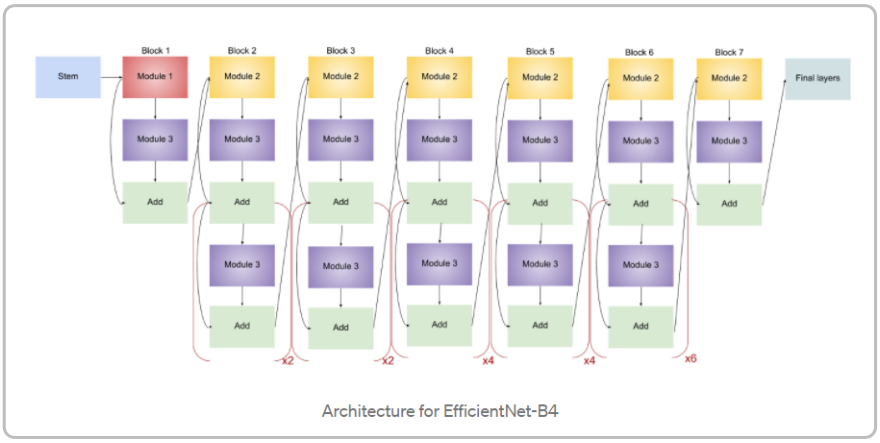
YOLO an acronym for 'You only look once', is an object detection algorithm that divides images into a grid system. Each cell in the grid is responsible for detecting objects within itself. YOLO is one of the most famous object detection algorithms due to its speed and accuracy.

This object detection model has been trained on CrowdHuman Dataset



Efficient-Net

Efficient-Net is a convolutional neural network architecture and scaling method that uniformly scales all dimensions of depth/width/resolution using a compound coefficient. Unlike conventional practice that arbitrary scales these factors, the Efficient-Net scaling method uniformly scales network width, depth, and resolution with a set of fixed scaling coefficients.



Architecture for EfficientNet-B4

Datasets Used

PETA Dataset

PETA Dataset is a dataset for recognizing pedestrian attributes. This dataset is by far the largest of its kind covering more than 60 attributes on 19000 images. It contains the images of pedestrians which are obtained from CCTV Surveillance. Images in this dataset clearly capture the body of the person rather than their faces so, we have used this dataset for predicting the age and gender of a person through their body.

UTKFace

The UTKFace dataset is a large-scale face dataset with a long age span (range from 0 to 116 years old). The dataset consists of 23,708 face images with annotations of age, gender, and ethnicity. The images cover large variations in pose, facial expression, illumination, occlusion, resolution, etc. For the training purpose, we have removed images with an age greater than 80.

We have used this dataset to predict the age and gender of the person through their face.

CrowdHuman Dataset

We have used this dataset to fine-tune the YOLOv5 model to obtain better results at the task at hand.

CrowdHuman is a benchmark dataset to better evaluate detectors in crowd scenarios. The CrowdHuman dataset is large, rich-annotated, and contains high diversity. CrowdHuman contains 15000, 4370, and 5000 images for training, validation, and testing, respectively. There are a total of 470K human instances from train and validation subsets and 23 persons per image, with various kinds of occlusions in the dataset.

Thank You