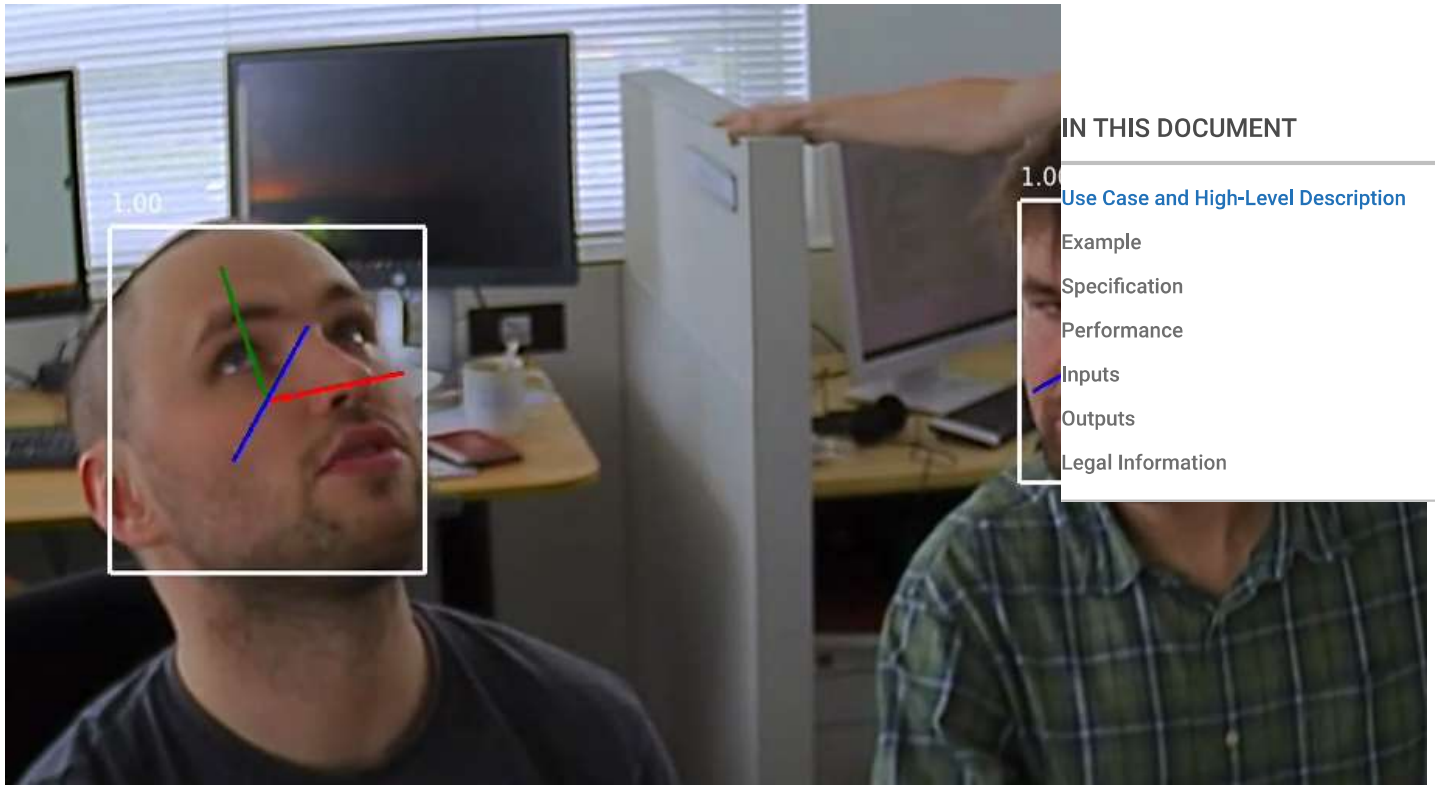
face-detection-adas-binary-0001

IN THIS DOCUMENT

Use Case and High-Level Description

Face detector for driver monitoring and similar scenarios. The network features a pr
includes depth-wise convolutions to reduce the amount of computation for the 3x3 c
1x1 convolutions are binary that can be implemented using effective binary XNOR+P

Example



IN THIS DOCUMENT

Specification

METRIC VALUE

- Samples
- DL Streamer Samples
- Demos
- Tools
- Intel Pre-Trained Models
  - Object Detection Models
    - faster-rcnn-resnet101-coco-sparse-60-0001
    - face-detection-adas-0001
    - face-detection-adas-binary-0001
    - face-detection-retail-0004
    - face-detection-retail-0005
    - face-detection-0100

METRIC	VALUE
AP (head height >10px)	31.2%
AP (head height >32px)	76.2%
AP (head height >64px)	90.3%
AP (head height >100px)	91.9%
Min head size	90x90 pixels on 1080p
GFlops	0.611
GI1ops	2.224
MParams	1.053
Source framework	PyTorch*

Average Precision (AP) is defined as an area under the precision/recall curve. Numbers are on Wider Face validation subset.

Performance

Inputs

Name: `input`, shape: [1x3x384x672] - An input image in the format [BxCxHxW], where:

- B - batch size
- C - number of channels
- H - image height
- W - image width

Expected color order is BGR.

Outputs

The net outputs blob with shape: [1, 1, N, 7], where N is the number of detected bounding boxes. Each detection has the format [`image_id`, `label`, `conf`, `x_min`, `y_min`, `x_max`, `y_max`], where:

- `image_id` - ID of the image in the batch
- `label` - predicted class ID
- `conf` - confidence for the predicted class
- (`x_min`, `y_min`) - coordinates of the top left bounding box corner
- (`x_max`, `y_max`) - coordinates of the bottom right bounding box corner.

Legal Information

[*] Other names and brands may be claimed as the property of others.

The NET was tuned from face-detection-adas-0001 weights