Models and Accuracies

Note: Kenneth Jung noticed that the model definitions are slightly different than the pre-trained models. For more information, see issues #351 (https://github.com/cmusatyalab/openface/issues/351) and #349 (https://github.com/cmusatyalab/openface/issues/349).

This page overviews different OpenFace neural network models and is intended for advanced users.

Model Definitions

The number of parameters are with 128-dimensional embeddings and do not include the batch normalization running means and variances.

Model	Number of Parameters
nn4.small2 (https://github.com/cmusatyalab/openface/blob/master/models/openface/nn4.small2.def.lua)	3733968
nn4.small1 (https://github.com/cmusatyalab/openface/blob/master/models/openface/nn4.small1.def.lua)	5579520
nn4 (https://github.com/cmusatyalab/openface/blob/master/models/openface/nn4.def.lua)	6959088
nn2 (https://github.com/cmusatyalab/openface/blob/master/models/openface/nn2.def.lua)	7472144

Pre-trained Models

Models can be trained in different ways with different datasets. Pre-trained models are versioned and should be released with a corresponding model definition. Switch between models with caution because the embeddings not compatible with each other.

The current models are trained with a combination of the two largest (of August 2015) publicly-available face recognition datasets based on names: FaceScrub (http://vintage.winklerbros.net/facescrub.html) and CASIA-WebFace (http://arxiv.org/abs/1411.7923).

The models can be downloaded from our storage servers:

- nn4.v1 (https://storage.cmusatyalab.org/openface-models/nn4.v1.t7)
- nn4.v2 (https://storage.cmusatyalab.org/openface-models/nn4.v2.t7)
- nn4.small1.v1 (https://storage.cmusatyalab.org/openface-models/nn4.small1.v1.t7)
- nn4.small2.v1 (https://storage.cmusatyalab.org/openface-models/nn4.small2.v1.t7)

API differences between the models are:

Model	alignment `landmarkIndices`
nn4.v1	`openface.AlignDlib.INNER_EYES_AND_BOTTOM_LIP`
nn4.v2	`openface.AlignDlib.OUTER_EYES_AND_NOSE`

Model	alignment `landmarkIndices`
nn4.small1.v1	`openface.AlignDlib.OUTER_EYES_AND_NOSE`
nn4.small2.v1	`openface.AlignDlib.OUTER_EYES_AND_NOSE`

Performance

The performance is measured by averaging 500 forward passes with util/profile-network.lua (https://github.com/cmusatyalab/openface/blob/master/util/profile-network.lua) and the following results use OpenBLAS on an 8 core 3.70 GHz CPU and a Tesla K40 GPU.

Model	Runtime (CPU)	Runtime (GPU)
nn4.v1	75.67 ms ± 19.97 ms	21.96 ms ± 6.71 ms
nn4.v2	82.74 ms ± 19.96 ms	20.82 ms ± 6.03 ms
nn4.small1.v1	69.58 ms ± 16.17 ms	15.90 ms ± 5.18 ms
nn4.small2.v1	58.9 ms ± 15.36 ms	13.72 ms ± 4.64 ms

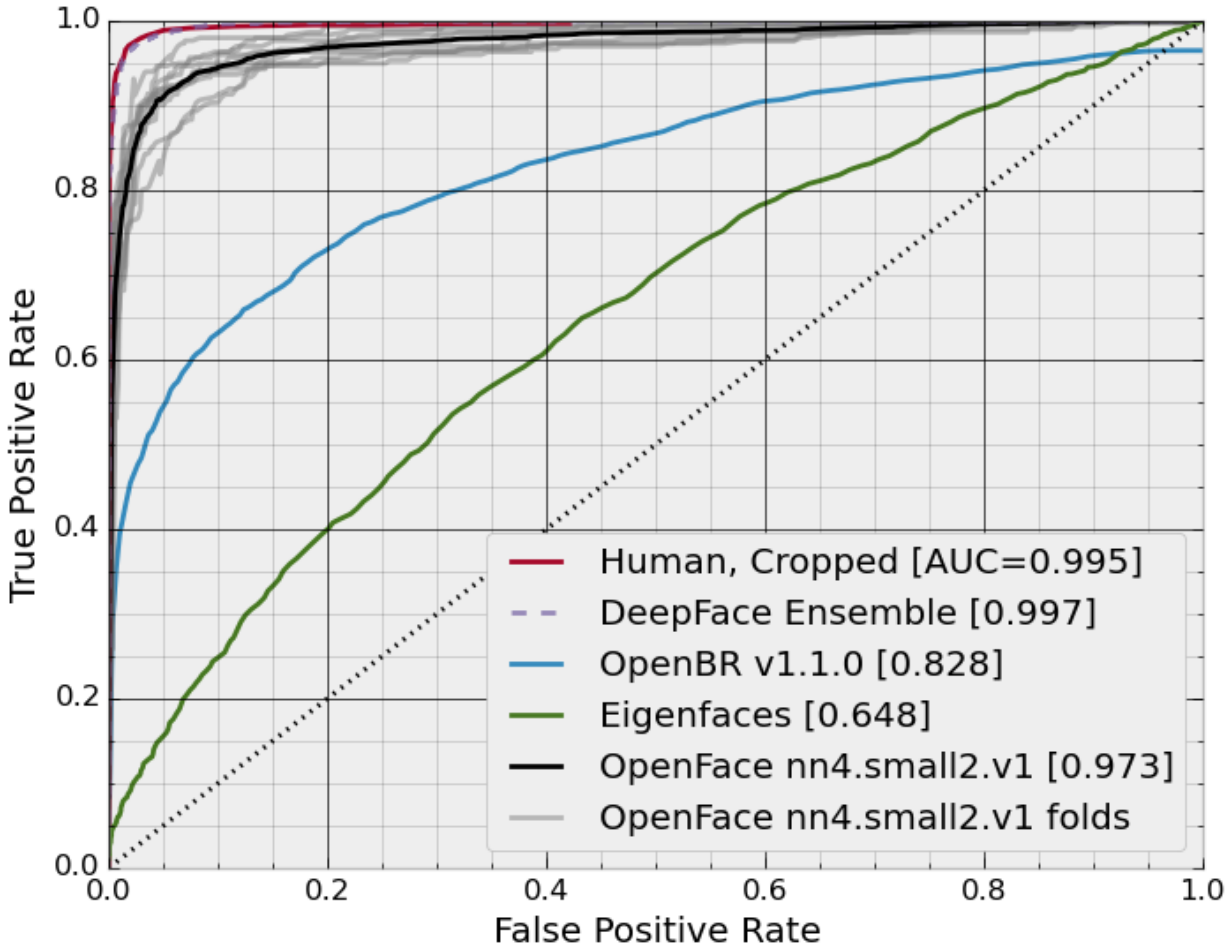
Accuracy on the LFW Benchmark

Even though the public datasets we trained on have orders of magnitude less data than private industry datasets, the accuracy is remarkably high on the standard LFW (http://vis-www.cs.umass.edu/lfw/results.html) benchmark. We had to fallback to using the deep funneled versions for 58 of 13233 images because dlib failed to detect a face or landmarks.

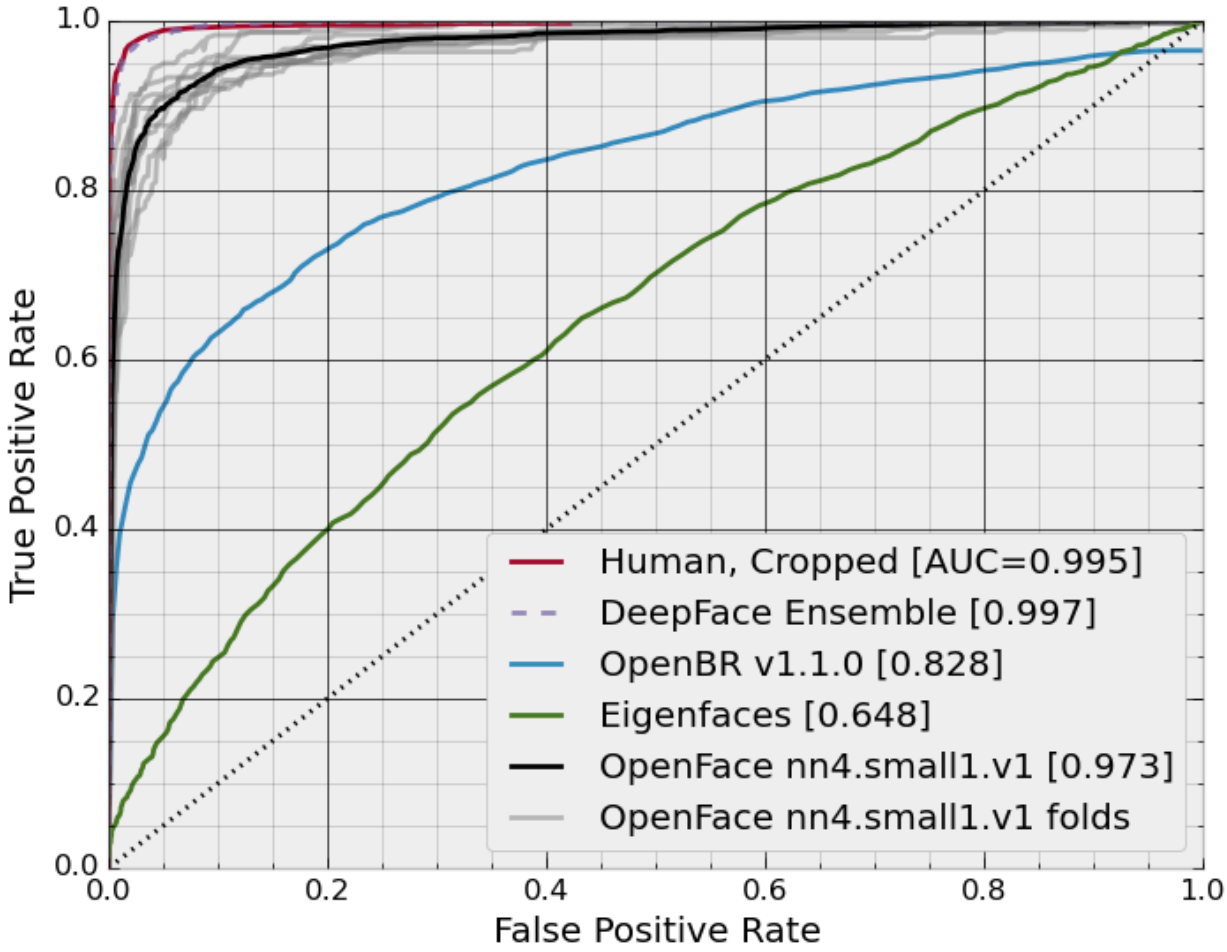
Model	Accuracy	AUC
nn4.small2.v1 (Default)	0.9292 ± 0.0134	0.973
nn4.small1.v1	0.9210 ± 0.0160	0.973
nn4.v2	0.9157 ± 0.0152	0.966
nn4.v1	0.7612 ± 0.0189	0.853
FaceNet Paper (Reference)	0.9963 ± 0.009	not provided

ROC Curves

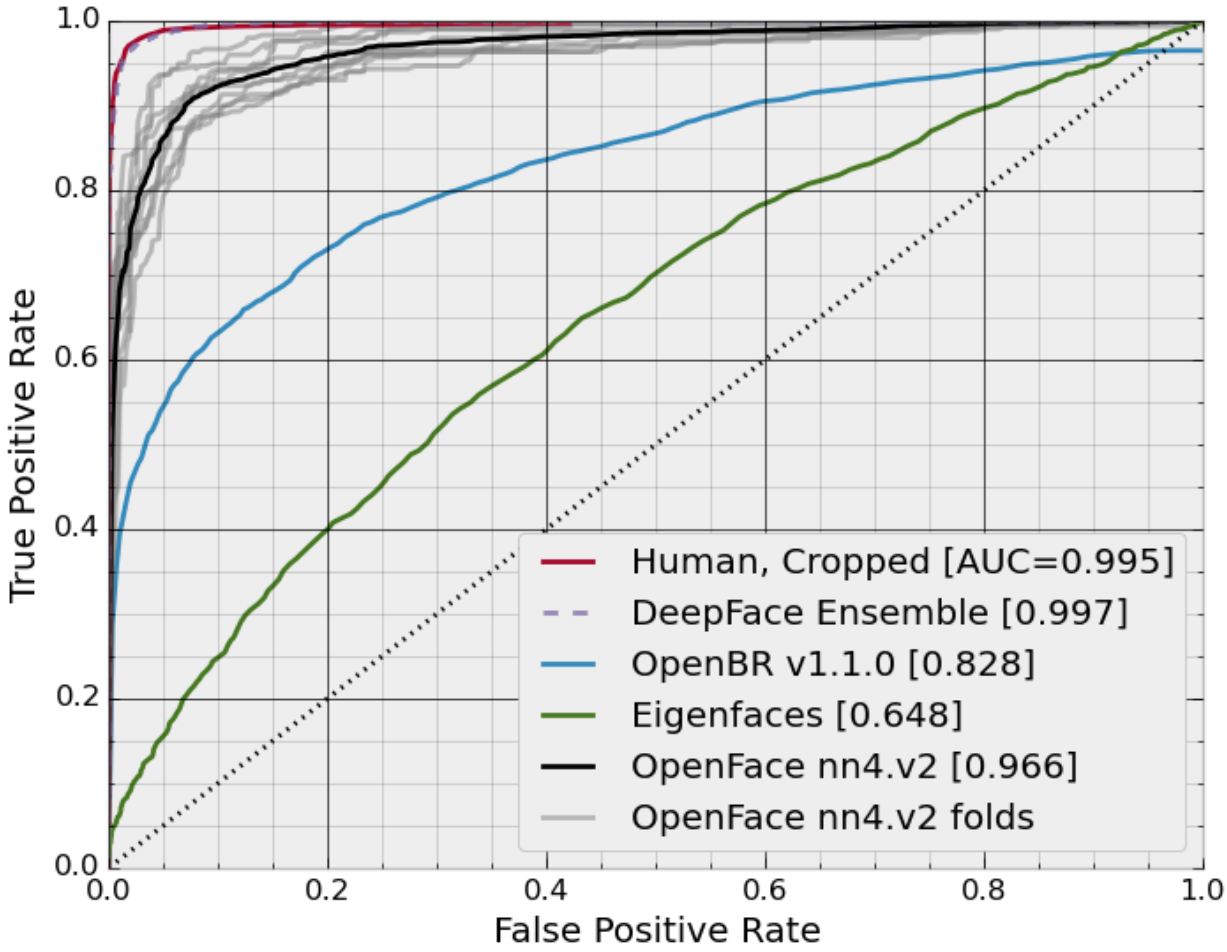
nn4.small2.v1



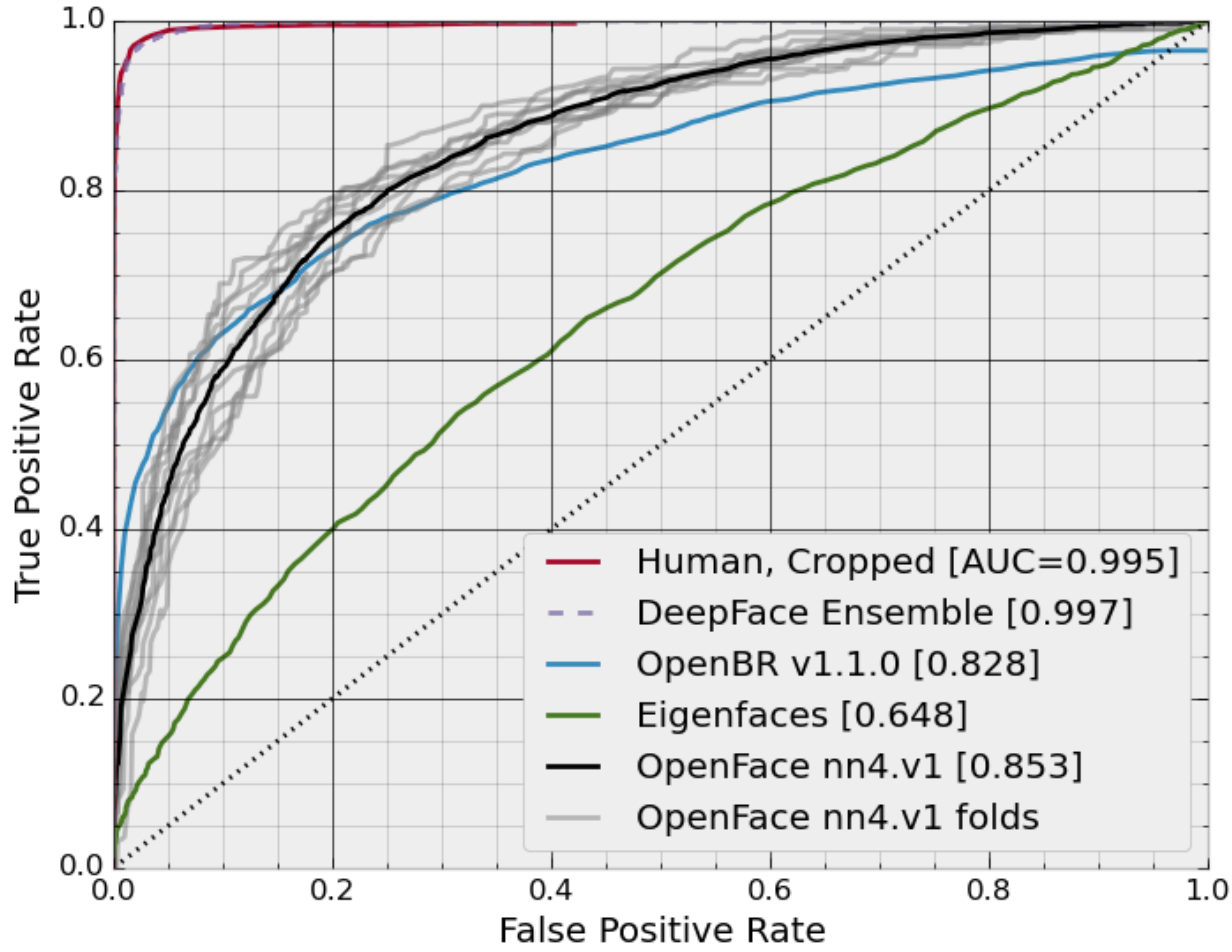
nn4.small1.v1


True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Human, Cropped [AUC=0.995]
DeepFace Ensemble [0.997]
OpenBR v1.1.0 [0.828]
Eigenfaces [0.648]
OpenFace nn4.small1.v1 [0.973]
OpenFace nn4.small1.v1 folds

nn4.v2


True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Human, Cropped [AUC=0.995]
DeepFace Ensemble [0.997]
OpenBR v1.1.0 [0.828]
Eigenfaces [0.648]
OpenFace nn4.v2 [0.966]
OpenFace nn4.v2 folds

nn4.v1



Running The LFW Experiment

This can be generated with the following commands from the root `openface` directory, assuming you have downloaded and placed the raw and deep funneled (http://vis-www.cs.umass.edu/deep_funnel.html) LFW data from here (http://vis-www.cs.umass.edu/lfw/) in `./data/lfw/raw` and `./data/lfw/deepfunneled`. Also save pairs.txt (http://vis-www.cs.umass.edu/lfw/pairs.txt) in `./data/lfw/pairs.txt`.

1. Install prerequisites as below.
2. Preprocess the raw `lfw` images, change `8` to however many separate processes you want to run: `for N in {1..8}; do ./util/align-dlib.py data/lfw/raw align outerEyesAndNose data/lfw/dlib-affine-sz:96 --size 96 & done`. Fallback to deep funneled versions for images that dlib failed to align: `./util/align-dlib.py data/lfw/raw align outerEyesAndNose data/lfw/dlib-affine-sz:96 --size 96 --fallbackLfw data/lfw/deepfunneled`
3. Generate representations with `./batch-represent/main.lua -outDir evaluation/lfw.nn4.small2.v1.reps -model models/openface/nn4.small2.v1.t7 -data data/lfw/dlib-affine-sz:96`
4. Generate the ROC curve from the `evaluation` directory with `./lfw.py nn4.small2.v1 lfw.nn4.small2.v1.reps`. This creates `roc.pdf` in the `lfw.nn4.small2.v1.reps` directory.

Projects with Higher Accuracy

If you're interested in higher accuracy open source code, see:

Oxford's VGG Face Descriptor (http://www.robots.ox.ac.uk/~vgg/software/vgg_face/)

This is licensed for non-commercial research purposes. They've released their softmax network, which obtains .9727 accuracy on the LFW and will release their triplet network (0.9913 accuracy) and data soon (?).

Their softmax model doesn't embed features like FaceNet, which makes tasks like classification and clustering more difficult. Their triplet model hasn't yet been released, but will provide embeddings similar to FaceNet. The triplet model will be supported by OpenFace once it's released.

Deep Face Representation (https://github.com/AlfredXiangWu/face_verification_experiment)

This uses Caffe and doesn't yet have a license. The accuracy on the LFW is .9777. This model doesn't embed features like FaceNet, which makes tasks like classification and clustering more difficult.
