DeepFace: Closing the Gap to Human-Level Performance in Face Verification

1 Citation

Taigman, Yaniv, et al. "Deepface: Closing the gap to human-level performance in face verification." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

`https://www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Taigman_DeepFace_Closing_the_2014_CVPR_paper.pdf`

2 Abstract

Face recognition has 4 steps: detect, align, represent, classify. We improve alignment by using 3D face models. We improve representation with a 9-layer neural network with convolutional and locally connected layers. We get 97.35% accuracy on the Labeled Faces in the Wild dataset.

3 Introduction

Our system uses deep learning to extract features. It depends on our 3D model based alignment step.

4 Related Work

Neural networks have become popular as datasets have gotten larger and we've figured out how to train them on GPUs.

State of the art face recognition are sensitive to lighting, occlusion, and more. Systems use tens of thousands of image descriptors.

5 Face Alignment

Given an unconstrained image in the wild, we need to align the face so it takes up the full frame. Existing methods use a 3D face model, search for similar fiducial points from another dataset, or use an unsupervised technique. We think 3D models are the best idea.

First, we extract LBP histogram features. Then we use an Support Vector Regressor (SVR) designed to look for fiducial points. Once we find them, we can transform the image with the similarity matrix and do this all over again. We keep repeating until we localize the face well.

With the crop extracted from above, we then find six fiducial points and use them to scale, rotate, and translate the the crop into a better 2D alignment. This doesn't handle out of plane rotation of the face (e.g. going from facing the camera to showing your profile is an out-of-plane rotation).

The next step is to identify 67 fiducial points (with an SVR) and map the face to a 3D model (computed from the USF Human-ID database). Then, we figure out the camera pose and align the face accordingly. Since the camera pose estimate is imperfect, we have some steps to clean it up.

6 Representation

Face recognition is multiclass classification. The input is a 3D aligned 152×152 RGB image.

The image goes into a 11×11 conv layer to yield 32 feature maps. Then we do 3×3 max pooling with stride 2. Next is a 9×9 conv layer that yields 16 feature maps. The goal of these layers is simply to extract some low level features.

The next three layers are locally connected layers. This means they apply feature maps, but the feature maps are different across the image. This makes sense because the eyes and nose area of the image is very different than the mouth and chin area, so you can't re-use a filter on both. Locally connected layers are just as computationally cheap as conv layers, but they have many more parameters.

Then, we have two fully connected layers. The output of the first FC layer can be use as a face descriptor.

Finally, we have a softmax layer. We use ReLU activation throughout.

We minimize cross entropy loss with stochastic gradient descent with Dropout.

Once we are done, we then take the output of the first FC layer as our descriptor $G(I)$. It tends to be pretty sparse. We normalize the features to lie in $[0, 1]$ to make them insensitive to lighting changes. To do this, we divide each element of the feature vector by the largest value of that element in the training set:

$$\bar{G}(I)_i = \frac{G(I)_i}{\max(G_i, \epsilon)}$$

Then, we L2 normalize $f(I) = \bar{G}(I)/||\bar{G}(I)||_2$

7 Verification Metric

Typically, you'd fine-tune to your new dataset, but we want our system to generalize to all datasets, so we won't.

To compute similarity of two faces, we just take the inner product of their descriptors. We also looked at two supervised metrics: χ^2 similarity and Siamese network.

$$\chi^2(f_1, f_2) = \sum_i w_i (f_1[i] - f_2[i])^2/(f_1[i] + f_2[i])$$

The weights w_i above are learned with an SVM and f represents a face descriptor.

The Siamese network looks like this. Each of the two input images has our neural network stacked on top of it (up to the part where we create feature descriptors). Then, we compute the absolute difference of the feature descriptors, send this through a fully connected layer, and then to a logistic unit that predicts whether the images show the same face. We fine-tune only the last two layers. The result is a distance metric:

$$d(f_1, f_2) = \sum_i \alpha_i |f_1[i] - f_2[i]|$$

where the α_i are learned by the network.

8 Experiments

We use the Social Face Classification Dataset (SFC) (collected from Facebook). We apply our descriptors to Labeled Faces in the Wild (LFW) and and YouTube Faces (YTF). SFC has 4.4M faces from 4K people. LFW has 13K faces of 5.7K celebrities. Our metric is mean recognition accuracy. YTF has 3.4K videos with 1.6K faces.

We train on SFC for 3 days with minibatch (128) gradient descent (learning rate 0.01) with momentum (0.9) for 15 epochs.

We find that our network doesn't saturate - larger dataset would help.

On LFW, removing the 3D alignment (i.e frontalization) gives 94.4%. Using no alignment gives 87.9%. Using frontalization and LBP/SVM instead of our neural network gives 91.4%. Our neural network with inner-product similarity gets 95.92% accuracy. Using the χ^2 metric gives 97% accuracy (human is 97.5%). Ensembling gives 97.15%. The Siamese network can't be fine-tuned directly on LFW or it will overfit, so we train on some additional data and do two epochs on LFW. This gives 97.25% (97.35% with ensembling). We set state of the art (previous was around 80%). We get 92.5% on YTF.

On a CPU, we can process 3 images a second.

9 Conclusion

An ideal face classifier would need to be invariant to pose, illumination, expression, and image quality. It should be generalizable. It should have short (preferably sparse) descriptors. It should be fast.

By combining 3D alignment with a neural network feature descriptor, we've achieved many of these goals and produce near-human face recognition accuracy.