Machine Learning is Fun Part 4

Yuan An

In the last three parts, machine learning is used to solve isolated problems, *e.g*. predicting house sale prices, generating new data based on existing data and recognizing objects, that have only one step. Now Iam learning something about Modern Face Recognition with Deep Learning according to the article written by Geitgey [2].

When we enter Qzone, we could notice that there is a box on the right side column. On the box, you can see a picture uploaded by your friends with face , an question that "He or she is your friend xxx?" and two buttons with "yes" or "no". It seems like the function Facebook released that has an ability to recognize face in the photograph. In the old days, Facebook used to make you to tag your friends in photos by clicking on them and typing in their name. Now as soon as you upload a photo, Facebook tags everyone in a magic style.

In part 4 of Geitgey's article [2], the author was talking about how modern face recognition works. He took telling Will Ferrell (famous actor) apart from Chad Smith (famous rock musician) as an example (see Fig. 1).

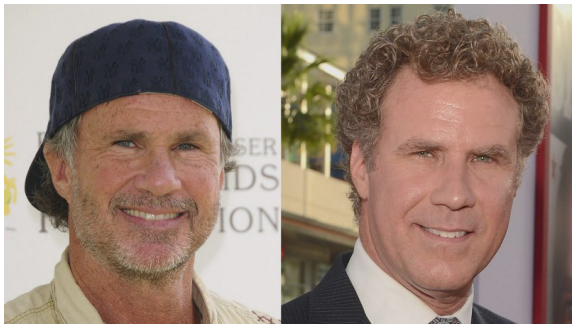

Figure 1. Chad Smith (left) and Will Ferrell (right)

Face recognition is really a series of several related problems:

1. Look at a picture and find all the faces in it.
2. Focus on each face and be able to understand that even if a face is turned in a weird direction or in bad lighting, it is still the same person.
3. Be able to pick out unique features of the face that you can use to tell it apart from other people–like how big the eyes are, how long the face is, etc.
4. Compare the unique features of that face to all the people you already know to determine the person's name.

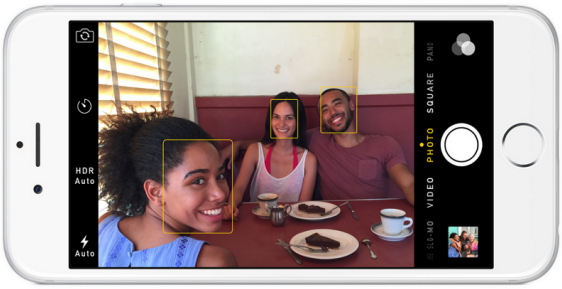

Figure 2. Face recognition in mobile phone.

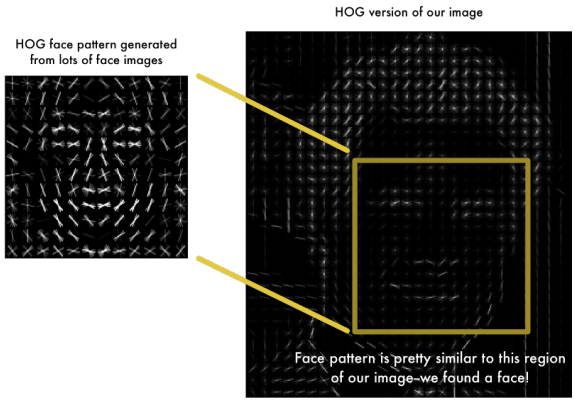

Figure 3. HOG image.

As a human, your brain is wired to do all of this automatically and instantly. But computers are not capable of this kind of high-level generalization. So what humans do is teaching computers how to do each step in this process separately.

The fist step in face recognition is face detection. We can find this function in our mobile phones (see in Fig. 2).

Face detection is a great feature for cameras. When the camera can automatically pick out faces, it can make sure that all faces are in focus before if takes the picture. It went mainstream in the early 2000's when Paul Viola and Michael Jones invented a way to detect face that was fast enough to run on cheap cameras. More about ViolaJones object

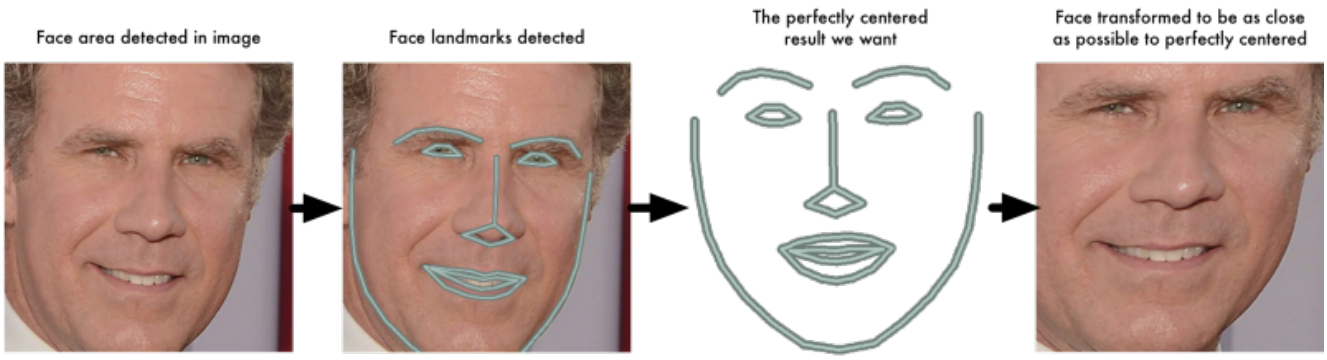

Figure 4. The face image after processing.

detection framework in [5, 3]. Now there are more reliable solutions. In the part 4 of the article, a method named Histogram of Oriented Gradients (HOG) is applied [1].

To find faces in an image, it should be converted to gray level image. Then processing it by several steps, the original image is turned into a HOG. And find the part of the image that looks the most similar to a kown HOG pattern that was extracted from a bunch of other training faces, as shown in Fig. 3.

The next step is posing and projecting faces. An isolated face was gotten in the last step. Now the problem should be deal with that faces turned different directions look totally different to a computer. So we will try to warp each picture so that the eyes and lips are always in the sample place in the image. After doing this, it is a lot easier for computer to compare faces in the next steps. To do this, an algorithm called face landmark estimation would be introduced. There are lots of ways to do this, the approach invented in 2014 by Vahid Kazemi and Josephine Sullivan [4] is going to used.

The basic idea of the approach is that there are 68 specific points (called landmarks) that exist on the human face (see Fig. 5). Then a machine learning algorithm will be trained to be able to find these 68 specific points on any face.

Now that we know were the eyes and mouth are, then rotate, scale and shear the image so that they eyes and mouth are centered as shown in Fig. 4.

The third step is Encoding faces. The simplest approach to face recognition is to directly compare the unknown with all the pictures. Then find the most similar face, and it must be the same person. But there's actually a huge problem with that approach. A site like Facebook with billions of users and a trillion photos can't possibly loop through every previous-tagged face to compare it to every newly uploaded picture. So extracting a few basic measurements from each face is a good way. Then measure unknown face that the same way and find the known face with the closest measurements.

Therefore, the solution is to train a deep convolutional neural network to generate 128 measurements for each face. The training process works by looking at 3 face images at a

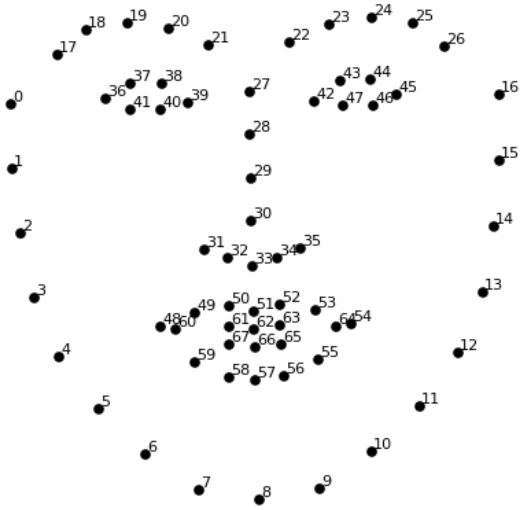

Figure 5. The 68 landmarks exist on the human face.

time:

1. Load a training face image of a known person.
2. Load another picture of the same known person.
3. Load a picture of a totally different person.

Then the algorithm looks at the measurements it is currently generateing for each of those three image. It then tweak the neural network slightly so that it makes sure the measurements it generates for #1 and #2 are slightly closer while making sure the measurements for #2 and #3 are slightly further apart (the processing seems like Fig. 6).

After repeating this step millions of times, the neural network learns to reliably generate 128 measurements for each person.

The final step is finding the person's name from the encoding. This is actually the easiest step in the whole process. So what we should do is using any basic machine learning classification algorithm, *e.g.* a linear SVM classifier.

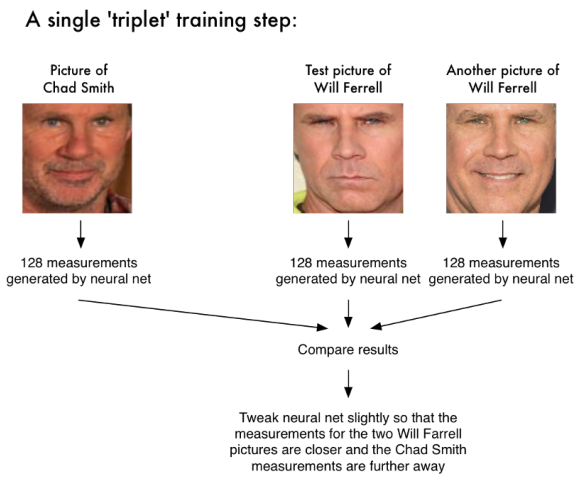

Figure 6. A single triplet training step.

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[2] A. Geitgey. Machine learning is fun. `https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471`.

[3] M. Jones and P. Viola. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):87–112, 2001.

[4] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *CVPR*, pages 1867–1874, 2014.

[5] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, pages 511–518, 2001.