Stress Testing Facial Recognition with Adversarial Examples

Emily Strong
CSYE 7245: Big Data Systems & Intelligence Analytics
February 9, 2018

TOPIC DESCRIPTION

In this project, I will stress test the OpenFace facial recognition algorithm to identify its susceptibility to adversarial examples, input designed to deliberately cause misclassification. I will particularly focus on adversarial input not detectable to the human eye. This will be achieved through the addition of noise or other minor changes to the images optimized through black-box attacks and a particle swarm optimization algorithm.

BACKGROUND

State-of-the-art facial recognition algorithms have met or surpassed human abilities (Facebook's DeepFace [1] has 97.25% accuracy and Google's FaceNet [2] has 99.63% accuracy, compared to human performance of 97.53%), however as with other classification algorithms, facial recognition models are susceptible to adversarial examples. Sharif et al [3] identified that facial recognition algorithms can be tricked using total-face pixel manipulation or texture perturbing eyeglass frames to either misidentify the face or impersonate a different face.

This result is intriguing as facial recognition algorithms encode distances of morphological features rather than individual pixels. Though the images typically need to be normalized before use, faces photographed at an angle or even a side view can still be correctly labeled, indicating that these measurements are robust to extreme distortions.

In their research, Sharif et al focused primarily on the frames as these have the potential for adversarial attacks in the real world. However, this method would not trick face recognition software without notice – anyone familiar with the glasses would recognize them in an image. I am thus interested in further exploring total-face pixel manipulation. Such attacks might be made against police and federal criminal databases, surveillance footage, or other systems used in security and criminal investigations. To mimic these types of scenarios, I will be using black-box attacks.

I will stress test the pixel manipulation method using a variety of test cases to determine sensitivity to conditions such as image exposure and degree of morphological differences between faces, as well whether images of non-human faces and images without faces at all can be used to trick the algorithm. Salah, Alyüz and Akarun [4] found that 3D scans of faces clustered based on morphological differences divide into clusters based on race and gender. The test cases for morphological differences will thus take

into account that race and gender are likely to correspond to greater morphological differences, examining each of these separately as well as combined.

While the most information would be gained from stress testing the highest scoring facial recognition algorithms, FaceNet is proprietary, as is DeepFace. I will thus be testing against the open-source OpenFace [5] which has 92.92% accuracy, very near that of human accuracy.

DATA SOURCES

I will be using the Labeled Faces in the Wild (LFW) data set [6]. It is available to directly import into a Python project as a SciKit-Learn module. The data set contains over 13,000 labeled pictures of famous people, with 1680 of the subjects having multiple photos in the set. I will supplement it with pictures of Natalie Portman from WikiMedia Commons as she is not in the data set but is essential to my second test case.

Data Sources: http://scikit-learn.org/stable/datasets/labeled_faces.html, https://commons.wikimedia.org/wiki/Natalie_Portman

ALGORITHMS AND CODE SOURCES

Face Detection
I will be using the dlib CNN Face Detector API and model, for which there is sample code available for getting started.

Code Sources: http://dlib.net/python/index.html#dlib.cnn_face_detection_model_v1, http://dlib.net/cnn_face_detector.py.html

Image Normalization/Affine Transformations
Since the LFW set includes faces that are not looking directly at the camera, the images must be normalized through a process known as affine transformation, in which parallel lines are preserved. I will be using OpenFace's AlignDlib API [11] which uses the dlib library to detect 68 facial landmarks that are then centered in the image and the image is resized. Figure 1 is an image from the API documentation indicating the landmarks.

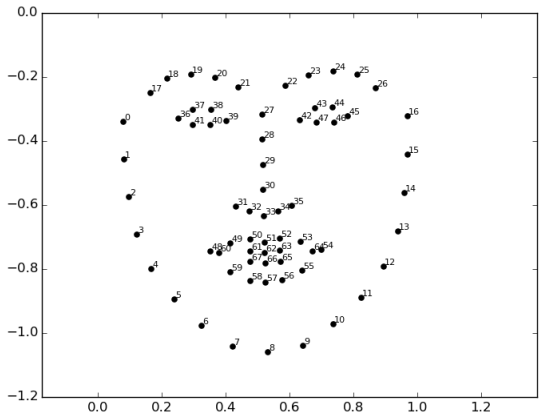

Fig 1: Facial Landmarks for Normalization

Code Source: http://openface-api.readthedocs.io/en/latest/openface.html#openface-alignDlib-class

Morphology Encoding
I will be using a pre-trained OpenFace model to encode the face morphology.

Code Source: https://github.com/cmusatyalab/openface/tree/master/models/openface

Morphology Classification ("Face Recognition")
I will use an SVM classifier on the encoding output to label the faces.

Code Source: http://scikit-learn.org/stable/modules/svm.html

Adversarial Image Manipulation
I will be adding noise to an image to confuse the facial detection, morphology encoding and morphology classification algorithms. There are several ways to do this as outlined in the answers to the below Stack Overflow post. I will use a Particle Swarm Optimization algorithm to rapidly generate potential solutions following a tutorial from SwarmIntelligence.org.

Code Sources: https://stackoverflow.com/questions/22937589/how-to-add-noise-gaussian-salt-and-pepper-etc-to-image-in-python-with-opencv/30609854, http://www.swarmintelligence.org/tutorials.php

STRESS TESTS

Each of the tests will be evaluated based on success in deceiving the algorithms, the number of solutions tested to achieve that result, and a qualitative assessment of whether the changes are detectable to the human eye. Additional test cases may be added.

* Indicates a supplement to the LFW data set

Case 1: Simple Image Changes
Subject: Keira Knightley
I will stress test with simple image manipulations such as lighting, color saturation, and noise to determine if these affect facial recognition as well as face detection.

Case 2: People with Similar Faces (Same Race, Same Gender)
Subjects: Keira Knightley, Natalie Portman*
Natalie Portman and Keira Knightley are frequently mistaken for each other. Knightly was famously cast in Star Wars: The Phantom Menace due to her resemblance to Portman and ability to stand in for Portman's character as a decoy in the plot [7]. I will manipulate images of each actress to determine if they can be modified through random pixel changes to be misidentified as each other.

Case 3: People with Dissimilar Faces (Same Race, Same Gender)
Subjects: Meryl Streep, Keira Knightley
I will repeat the process from Case 2 with an actress who does not morphologically resemble Keira Knightley – Meryl Streep.

Case 4: People with Different Genders
Subjects: Meryl Streep, Tom Hanks
I will then repeat the process with actors of different genders since gender is one of the clustering criteria for morphological differences determined by Salah et al [4]

Case 5: People of Different Races
Subjects: Morgan Freeman, Tom Hanks
I will repeat the process with actors of different races. This is particularly of interest due to the established inaccuracies of facial recognition algorithms in using the faces of people of color [8].

Case 6: People of Different Races and Genders
Subjects: Meryl Streep, Morgan Freeman
I will repeat the process with actors of different genders and races. As these are the two morphological clustering criteria, this will be the greatest stress test with human faces.

Case 7: Human Face and Non-Human Face
Subjects: Tom Hanks, a cat*
I will manipulate an image of a cat to determine if the face detection algorithm can be tricked into identifying a non-human face as a human face, and if so can that same image also trick the face encoding and classification algorithms into identifying that cat as one of the trained faces (Tom Hanks). Since none of the algorithms are trained to classify objects in images as cats I will only be modifying the cat image.

Case 8: Human Face and No Face
Subjects: Tom Hanks, a bicycle*
I will repeat the process from Case 7 with an image of an object that does not have a face (a bicycle) to see if it can be misidentified as a face, and specifically the face of Tom Hanks.

REFERENCES

[1] Taigman, Yaniv, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. "Deepface: Closing the gap to human-level performance in face verification." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1701-1708. 2014.

[2] Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815-823. 2015.

[3] Sharif, Mahmood, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition." In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1528-1540. ACM, 2016.

[4] Salah, Albert A., Nese Alyüz, and Lale Akarun. "Registration of three-dimensional face scans with average face models." *Journal of Electronic Imaging* 17, no. 1 (2008): 011006.

[5] Amos, Brandon, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. "Openface: A general-purpose face recognition library with mobile applications." *CMU School of Computer Science* (2016).

[6] Huang, Gary B., Manu Ramesh, Tamara Berg, and Erik Learned-Miller. "Labeled faces in the wild: A database for studying face recognition in unconstrained environments." *Vol. 1, no. 2. Technical Report 07-49*, University of Massachusetts, Amherst, 2007.

[7] Wikipedia. "Star Wars: Episode I – The Phantom Menace". https://en.wikipedia.org/wiki/Star_Wars:_Episode_I_%E2%80%93_The_Phantom_Menace. (retrieved 2/9/18)

[8] Garvie, Clare and Jonathan Frankle. "Facial-Recognition Software Might Have a Racial Bias Problem." *The Atlantic*, April 7, 2016. https://www.theatlantic.com/technology/archive/2016/04/the-underlying-bias-of-facial-recognition-systems/476991/. (retrieved 2/9/18)