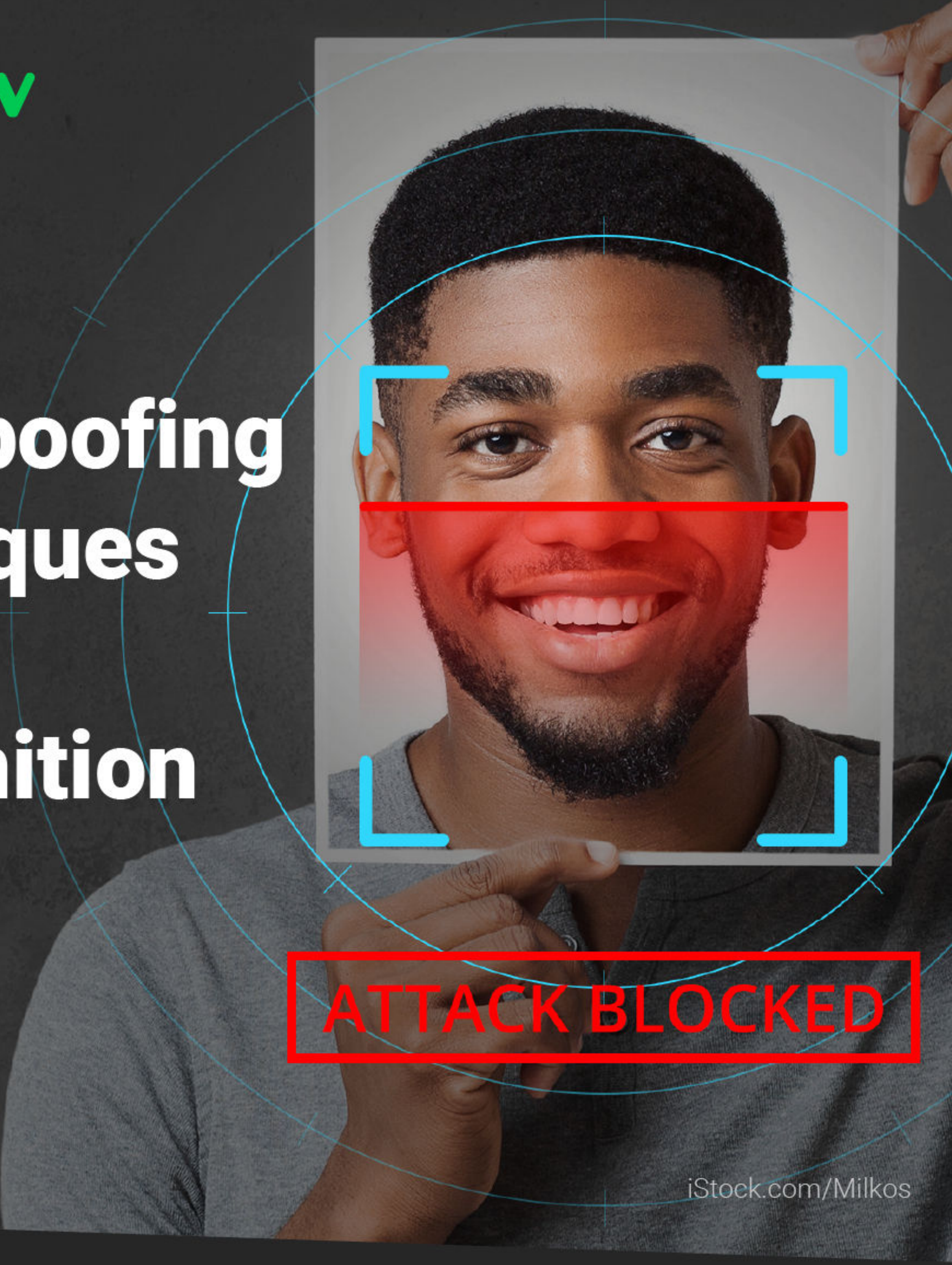
mobidev

Anti-Spoofing Techniques in Face Recognition



iStock.com/Milkos

written by

Sergey Maximenko

Data Science Solution Architect at MobiDev

mobidev

Upwork 2016 · Upwork 2017 · Upwork 2018

Best Upwork Software Development Agency In Ukraine

COMPLEX SOFTWARE DEVELOPMENT

WITH A FOCUS ON INNOVATION

WEB & CLOUD INFRASTRUCTURE

AR & MOBILE APPS

IOT & HARDWARE INTEGRATION

DATA SCIENCE & MACHINE LEARNING

For Startups

For Emerging companies

For Enterprises

Guaranteed delivery
on time and on budget
No surprises

You can **adapt to evolving business needs and increase ROI** with our flexible, proven processes

Top US-level quality for **1/3 the price** to bring **3x features** to your product

350+ Products launched

100% Approval rating by Upwork

300+ English speaking professionals

Find more at **www.mobidev.biz**

✉ info@mobidev.biz ☎ +1 888 380 0276

UNITED STATES OFFICE
Atlanta, Georgia

UNITED KINGDOM OFFICE
Sheffield

ENGINEERING OFFICES IN UKRAINE
Kharkiv, Chernivtsi, Mykolaiv

Table of Contents

Biometric face recognition technology is a key to security. Finding someone's photo or video on Facebook or Youtube is easy. These images and videos can be used for ill intent. Face-based biometric systems are vulnerable to attacks via paper photographs, screen replay or 3D face reconstruction. A security system designed to prevent face spoofing is important.

Following is an overview of presentation attacks and anti-spoofing techniques powered by Machine Learning.

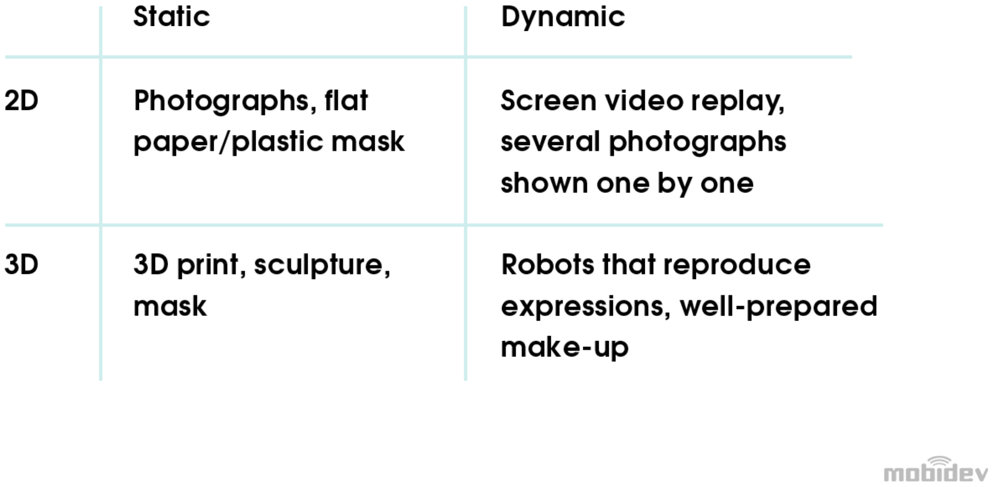
Face recognition: How it works

Face identification and recognition is a process of comparing data received from the camera to a database of known faces and finding the match. See the video below for an example:

This general face recognition process is flawed. What if someone uses a fake face? A liveness check counters this, distinguishing between a real face and a picture.

Presentation attacks: An overview

Attacks on a face recognition system are called Presentation Attacks, or PA. These attacks can be sorted into the following categories:



	Static	Dynamic
2D	Photographs, flat paper/plastic mask	Screen video replay, several photographs shown one by one
3D	3D print, sculpture, mask	Robots that reproduce expressions, well-prepared make-up

Of course, as technologies evolve, so do presentation attacks.

3D spoofing is not a big problem yet. 2D spoofing is more widespread. This puts the onus on detecting and preventing presentation attacks. The requirements are precise. The product must:

- combat 2D attacks, static or dynamic
- use images, not videos
- work without user interaction

The objective: achieve maximum accuracy in minimum time while also providing a user-friendly experience. A model meeting these requirements would be easy to integrate with existing face recognition systems.

This video provides an overview of face recognition spoofing:

PAD solutions: Top techniques

Presentation attack detection (PAD) technology stacks include:

Tensorflow: An open-source framework for building and computing data *flowgraphs, allowing for the creation and training of neural networks for any level of complexity*

Keras: High-level neural network application programming interface, or API, written in Python and capable of running on top of TensorFlow

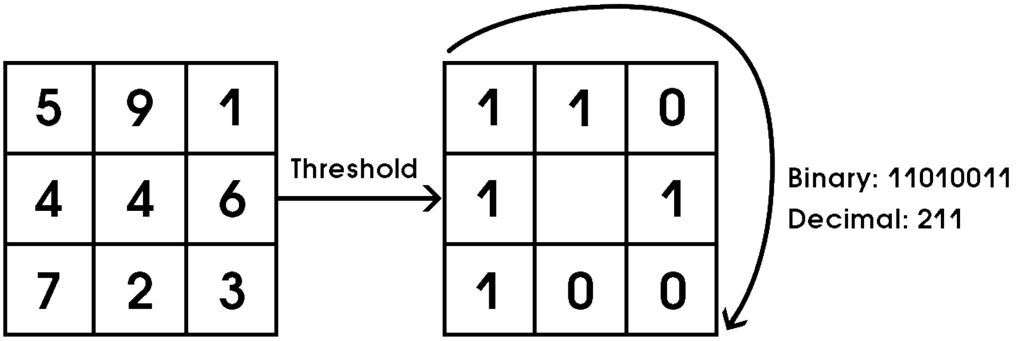
Scikit-learn: Software machine learning library for the Python programming language

OpenCV: An open-source computer vision and machine learning software library

During research PyTorch was also used, a machine learning library used for deep learning applications and natural language processing. While this framework has the same abilities as Tensorflow, its ease of use and flexibility make it popular among researchers. Following is a detailed overview of PAD solutions:

Hand crafted features: Local Binary Pattern

Local Binary Pattern (LBP) is an anti-spoofing technique of texture image analysis, thresholding image pixels according to their neighbors. An image is split into small patches. All pixels are taken from around the center pixel and subtracted. LBP is outdated and cannot be used against a large variety of attacks. But it does have merit when used in combination with other solutions.



5	9	1
4	4	6
7	2	3

1	1	0
1		1
1	0	0

Binary: 11010011
Decimal: 211

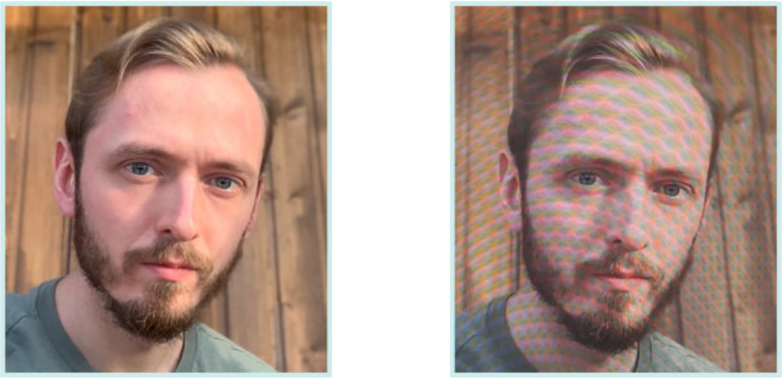
A negative result means 0 and a positive means 1. The resulting sequence of 0 and 1 (clockwise) is interpreted as a decimal number. The same calculations are made for all pixels in the patch. A histogram is constructed from the calculated values. This histogram is the texture descriptor of the patch. Histograms from all patches are then concatenated. That provides the feature descriptor of the image. It can be fed into a Support Vector Machine or any other algorithm, thus making a classification.

Finding the distinctions: Eye blink detection

Natural blinking can reveal the difference between live and fake faces. Humans blink an average of 15 to 30 times per minute. During a blink, the eyes remain shut for about 250 milliseconds. Modern cameras record videos with far smaller intervals between frames (50 milliseconds at 30 frames per second). Videos can be used to find frames showing the eyes shut. Those frames then can be counted. Implementation of eye blink detection can be based on face landmarks analysis and by calculating the area of the eye regions. Deep learning can also be applied for this task.

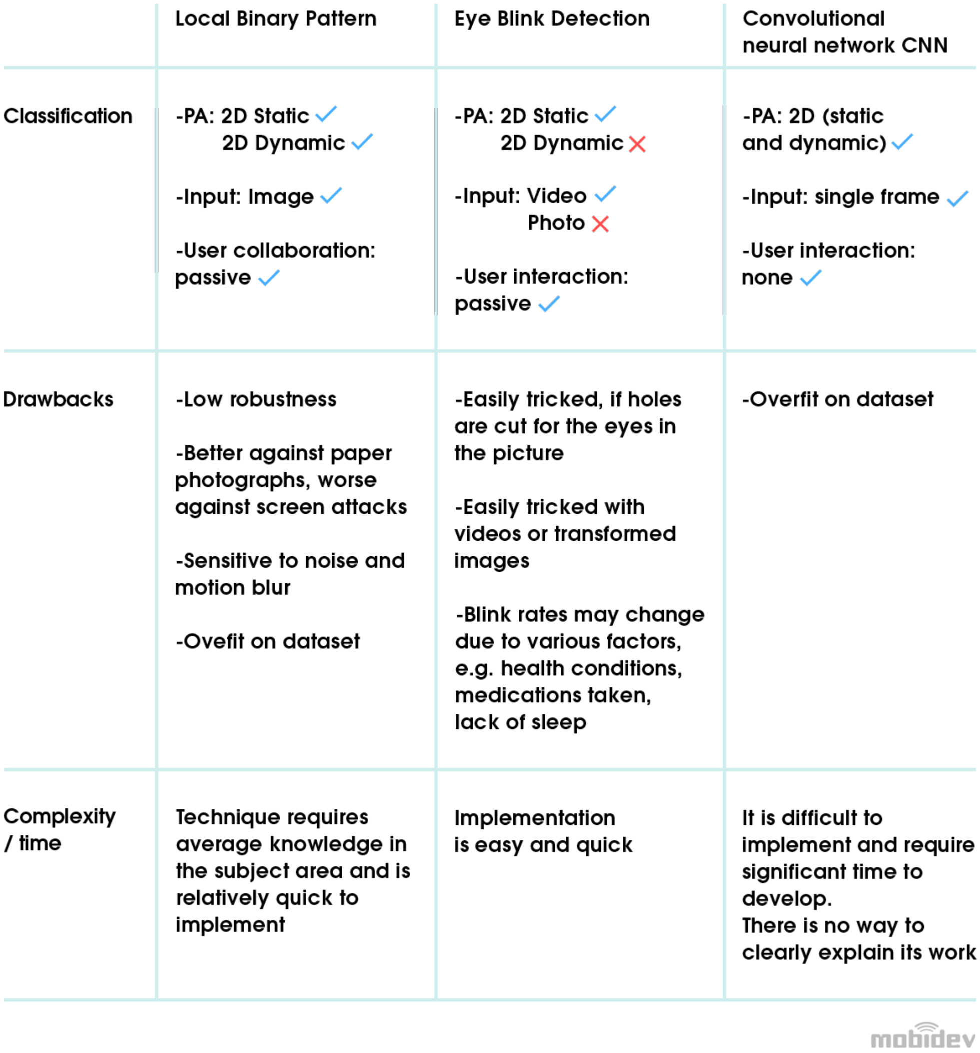
Deep learning features: Convolutional Neural Network

The development of convolutional neural networks (CNN) makes it natural to think of anti-spoofing as a binary classification problem. But there is no specific set of features the convolutional neural network would "see" and "understand." The hope is that trained convolution kernels will detect what human eyes can't see.



For example, distortions, if there are any, in the left picture can't be seen. Distortions are visible in the right picture, but they are so diverse and dependent on the environmental conditions and device specifics that the network can't use the image to generalize.

The network can work, but there's a high risk of overfit on the dataset. Face anti-spoofing performance will be strong within a certain dataset, but the network won't work in real conditions. A more viable solution is needed.



	Local Binary Pattern	Eye Blink Detection	Convolutional neural network CNN
Classification	-PA: 2D Static ✓ 2D Dynamic ✓ -Input: Image ✓ -User collaboration: passive ✓	-PA: 2D Static ✓ 2D Dynamic ✗ -Input: Video ✓ Photo ✗ -User interaction: passive ✓	-PA: 2D (static and dynamic) ✓ -Input: single frame ✓ -User interaction: none ✓
Drawbacks	-Low robustness -Better against paper photographs, worse against screen attacks -Sensitive to noise and motion blur -Ovefit on dataset	-Easily tricked, if holes are cut for the eyes in the picture -Easily tricked with videos or transformed images -Blink rates may change due to various factors, e.g. health conditions, medications taken, lack of sleep	-Overfit on dataset
Complexity / time	Technique requires average knowledge in the subject area and is relatively quick to implement	Implementation is easy and quick	It is difficult to implement and require significant time to develop. There is no way to clearly explain its work

*The most accurate result will be achieved through a combination of several solutions.

Comparing actions: Challenge-response

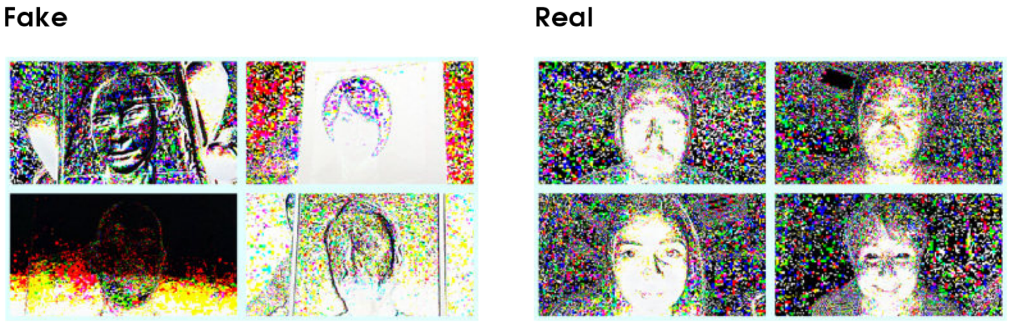
Under this face recognition approach, a user is required to take a special action called a challenge. The system ensures that required action was taken. Usually, a group of actions is required to make the model reliable. These actions can include smiles, expressions of emotions such as sadness, surprise or head movements. These interactions require significant time and are inconvenient for users.

Stable and reliable: 3D Camera

The most reliable anti-spoofing technique uses a 3D camera. Precise pixel depth information provides high accuracy against presentation attacks. The difference between a face and a flat shape is discernible. While 3D attacks still cause difficulties, stable performance makes this technology the most promising. It works with smartphones and web cameras. Technologies that work with RGB images are preferable.

An answer: Active Flash

This meets all the requirements. It has no "black box problem." It allows for the detection of spoofing using light reflections on a face. The idea is to change light environment by using the device screen as an additional source of light. White area, which covers the screen, produces appropriate reflection on the face.

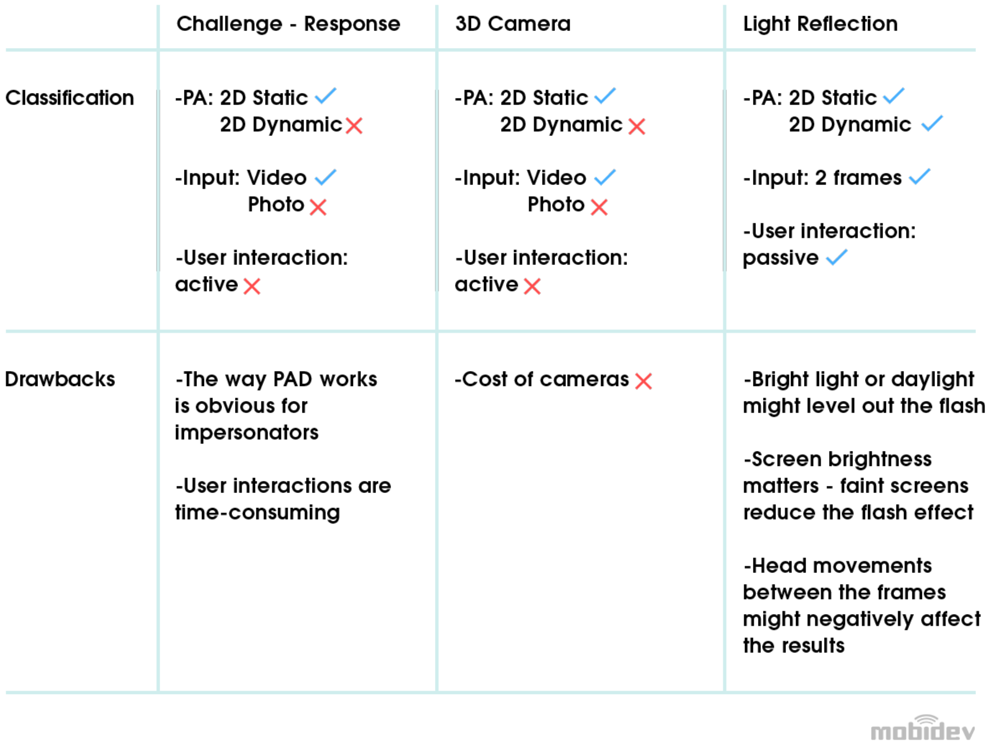


Fake

Real

Here is an example with raw pixel difference. In such an artificial way, real faces are distinguished from fake ones because of the differences in their surfaces. Frames before and after active flashing provide the data sample for training the network. Active flash helps separate and classify face features.

Note: *Although this model works regardless of how the head is turned (with reasonable limits), proper alignment can increase accuracy.*

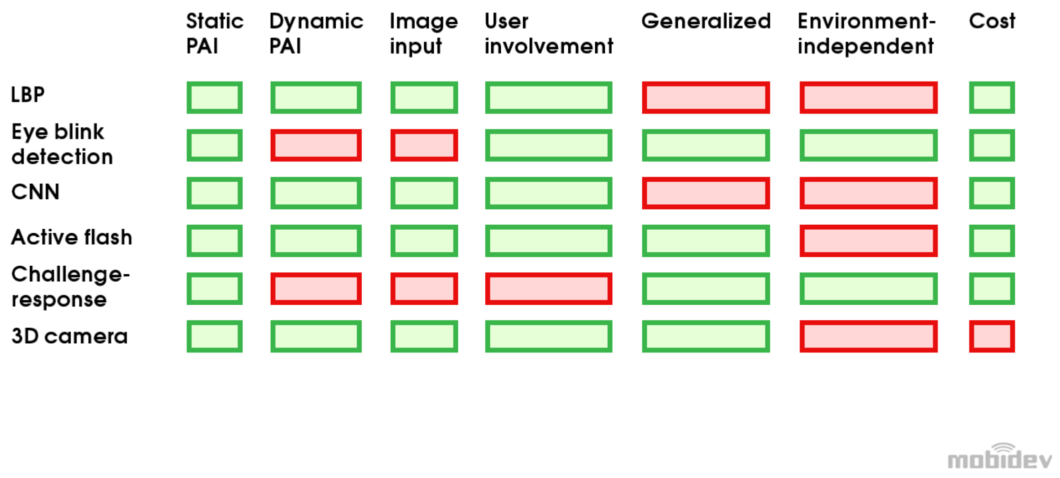


	Challenge - Response	3D Camera	Light Reflection
Classification	-PA: 2D Static ✓ 2D Dynamic ✗ -Input: Video ✓ Photo ✗ -User interaction: active ✗	-PA: 2D Static ✓ 2D Dynamic ✗ -Input: Video ✓ Photo ✗ -User interaction: active ✗	-PA: 2D Static ✓ 2D Dynamic ✓ -Input: 2 frames ✓ -User interaction: passive ✓
Drawbacks	-The way PAD works is obvious for impersonators -User interactions are time-consuming	-Cost of cameras ✗	-Bright light or daylight might level out the flash -Screen brightness matters - faint screens reduce the flash effect -Head movements between the frames might negatively affect the results

*Each of these solutions is difficult to implement and takes time to develop.The most accurate result will be achieved through a combination of solutions.

Techniques can be used both separately and in combination. Some examples:

Overview Of Face Anti-spoofing Solutions



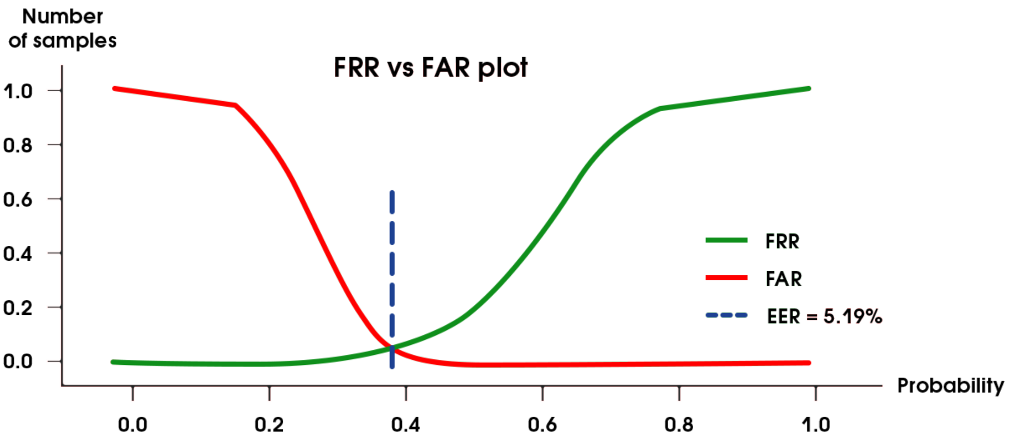
	Static PAI	Dynamic PAI	Image input	User involvement	Generalized	Environment-independent	Cost
LBP	🟩	🟩	🟩	🟩	🟥	🟥	🟩
Eye blink detection	🟩	🟥	🟥	🟩	🟩	🟩	🟩
CNN	🟩	🟩	🟩	🟩	🟥	🟥	🟩
Active flash	🟩	🟩	🟩	🟩	🟩	🟥	🟩
Challenge-response	🟩	🟥	🟥	🟥	🟩	🟩	🟩
3D camera	🟩	🟩	🟩	🟩	🟩	🟥	🟥

mobidev

Approaches can be combined. Challenge-response and LBP is one example. The first approach uses movements to determine face liveness. The second ensures these movements are not shown on a flat screen. Light reflection and challenge-response can be combined, showing whether a face appears on a flat surface and whether it's a static mask or 3D print.

Measuring efficacy: Metrics

The accuracy of an anti-spoofing system can and should be measured.



False Acceptance Rate (FAR) and False Rejection Rate (FRR), which are common for biometric verification, are also used for anti-spoofing. Task specifics define the metrics used to interpret errors. If the objective is to catch every attack, FAR should be minimized. If a user-friendly experience has higher priority, FRR becomes more important. If we visualize probabilities in the following way we will see that 2 curves intersect at a certain point. This point is Equal Error Rate and it helps to select the best threshold value for decision making. Depending upon the security requirements, we can move threshold value left or right giving an advantage to FAR or FRR. In our case, smooth user experience was more important so we adjusted the threshold value in order to minimize FRR.

Preventing attacks: What's next?

The question of presentation attack detection is still open in the research community. But there is opportunity for change.

Cheaper 3D cameras

It makes sense to use cheap hardware, if it can provide depth information, to get the safest and most reliable PAD system. A significant amount of recent research is based on datasets collected with the help of 3D cameras.

Anomaly detection

In each PAD approach mentioned above, the task was considered a two-class classification. Some propose that anomaly detection is better suited to anti-spoofing, making the network more generalized. All genuine samples have the same nature in common. Attack samples differ but are diverse and rarely combined into a single class.

Metric learning techniques

Metric learning techniques are applied to achieve better generalization by minimizing the variance in feature distributions. Variance can be caused by domain or identity specifics. Presentation attack instruments are considered as domains in this case. For example, print and screen replay attacks form separate distributions in the feature space. When the model faces new, unseen devices used for making an attack, there will be a new domain distribution and thus feature separation might lead to misclassifications. This problem can be solved by using distance-based metrics to help change the network training process to reduce separation between distributions.

Anti-spoofing based on deep learning is not hype. It's reality when well-defined product goals are combined with the right approaches and metrics.

mobidev

Your Software Development Partner

CONTACT US

info@mobidev.biz

mobidev.biz