Multi-layered Dictionary learning for Face Spoofing Detection

Atif Ahmed
Dept. of Computer Science
Columbia University
New York, NY 10027
{atif.ahmed}

Srinidhi Raghavan
Dept. of Computer Science
Columbia University
New York, NY 10027
{srinidhi.raghavan}
@columbia.edu

Thejaswi Muniyappa
Dept. of Computer Science
Columbia University
New York, NY 10027
{m.thejaswi}

Abstract

Automatic face recognition is becoming popular in many application areas such as bio-metric authentication systems. This wide usage has exposed many vulnerabilities of these systems. In particular, with face spoofing attacks these systems are bypassed by using photos or other artificial reconstructions of a face. The publicly available photos of people on online social networks has made it easier to obtain photos of peoples faces. In this paper, we use Multi-layered dictionary learning to learn a sparse representation of face images. These representations are then used to classify the images as to whether they are real faces or photographs, which is a difficult problem since both pictures will be very similar to each other. Further, we also evaluate our algorithm on the MNIST data-set and show how using a Multi-layered dictionary gives significant improvements in both time taken to train the classifiers, as well as the accuracy of different classifiers.

1 Introduction

The world of cinematography has become more exciting by the maturation of latex masks (commonly known as Silicone Masks). These masks act as a catalyst to bring about ways to obfuscate existing identities and build new ones. From the early 1990s movies like Mrs.Doubtfire to the involving film series like Harry Potter, masks have proven to be an integral identity. They make it possible to create realistic faces and give flexibility to come about all the possible facial expressions. However, it is high time to ponder upon the question Is it possible to ploy using these tools? Not to the utter surprise, the comeback is a yes. With the wide-ranged affordable availability of these masks, it becomes possible for the felonious fraudsters to use it for all sorts of impermissible crimes. One of such crimes happened in New York where the robbers used $2000 masks to fool $2,00,000 from the victims. The increasing outbursts of such crimes makes the addressal of face spoofing, the need of the hour.

1.1 Motivation

In the field of security, face recognition always stands out with favorable reconciliation between convenience and reliability. Unfortunately, due to its exceptional ease of access, face recognition is exposed to more serious threats among all the bio-metric traits like fingerprint, iris, DNA, retina, palm print, etc. This can be attributed to the facile access to the face samples, which in turn leads to various kinds of spoofing attacks like the ones discussed earlier. Despite the advancement in technology, vulnerability to such spoofing attacks is a serious drawback of the bio-metric systems. Therefore, it is essential to secure these systems against such manned presentation attacks.

Spoofing attacks does not restrict to silicone masks. Imposters find other ways to penetrate by using printed photographs, replaying recorded videos on mobile devices or by tying forged photographed masks. The consequences of these issues, the lack of proper infrastructure for their prevention motivates us to come with a way to detect spoofs. In this paper we aim to detect if a given image is a photograph of the face or a real face.

The rest of the paper is organized as follows: In Section 2, we discuss the existing techniques in this field. In Section 3, we describe the problem statement. We describe the architecture of the system in section 4. In Sections 5-7, the experimental set-up and the results are elaborated.

2 Literature Survey

One of the prominent initial works on face recognition was put forth in 2004 [5]. The growing popularity of facial recognition has attracted significant attention there-after. According to the prompts from various resources, it could be inferred that face spoof detection can be classified into four groups: (i) motion based, (ii) texture based, (iii) image quality based and (iv) based on background information. Motion based techniques work on printed photo attacks primarily and lack in both robustness and speed. Texture based are widely used for replay video attacks. Texture based techniques are better than the former counter-part, however, they lack in generalization. In image-quality based methods, the quality of the image is analyzed to conclude if it is a spoof or not. Lastly, in the ones based on background information, details like 2D intensity image are captured and worked upon. In the latter techniques, additional pre-processing is required, there-by making it slow. Though there are many existing techniques in this field, none of them generalize to find all types of spoofs or extend to find spoofs in complex environments. [6]

Presentation Attack Detection (PAD) algorithms were a triumphant attempt on face and iris biometrics [7]. Given a unobserved sample, the algorithm extracts statistical features like BSIF and 2D Cepstrum. All the extracted features are concatenated to form one single feature vector. The modified feature vector is then classified using a linear classifier. Despite its simplicity, these algorithms work effectively on photographed attacks. The more complex the features, the better is the performance

Principally, PAD algorithms modify the base feature vector to incorporate high-precision data. An equivalent technique involves encoding the temporal and spatial features of an image [8]. This couple with sparse encoding, makes it possible to make appropriate predictions even with minimal training data.

The advantages of PAD and sparse representations were put together in [1] by Ishan Manjani et. al. In this proposed work, deep layered dictionaries were used on a self-developed database known as SMAD (Silicone Mask-Attack Database). An efficient layered greedy approach is formulated to learn the dictionaries and then SVM is used as the classifier. From this paper, it is evident that, Dictionary Learning is applicable to these applications to a great extent. The way the dictionaries are learnt makes the difference here.

In this work, multi-layered dictionaries as mentioned in [1] is implemented. It is then compared with other techniques and is discussed why it has an upper hand. Moreover, its performance is analyzed by tuning various parameters. The following sections delve deeper into this.

3 Problem Statement

The implemented work is for face spoofing detection. Given an image, a sparse vectorized analog of the image is computed. This is done using a Multi-layered Dictionary learning process. The dictionary gives a summarized information of the input image on all the dimensions and the sparse approximate gives a reduced dimensional sparse representation of the input. It is this sparse representation which is further used for classification. If the data is unobserved, then the sparse representation of the data is computed with the help of the learnt dictionary.

The sparse representation is a metaphor of the input in a different space. The advantage of having a sparse representation is that it has low latency even for large data-sets. Besides, when you use a dictionary, you get a cumulative informed summary of all the training data. This makes sure to reduce the bias. Any non-linear classifier can be used for the given scenario. A non-linear classifier is required as data might not be split orderly in an affine fashion. For the sake of accuracy, SVM is chosen. SVM being a wide-margin classifier ensures to reproduce the most precise outcomes. This classifier is then tested on the data-set using the learned dictionaries.

4 Architecture

The architecture of the implemented technique is given in Figure 2. The input is an image and the output is the label saying if it is spoofed or not. SVM is used for classification and a deep layered-dictionary is used to represent the input.

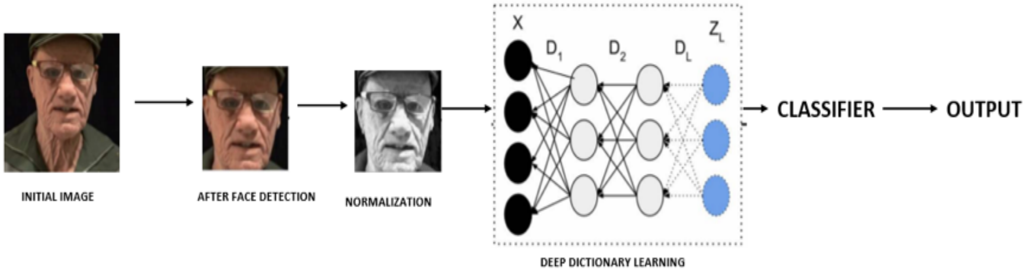


Figure 1: Architecture of the model

The step-by-step process are explained as follows:

1. Viola-Jones: In order to crop the face from the rest of the image, Viola-Jones [9] is used. It is a face detection framework, which uses AdaBoost for detecting the face. The detected face is then cropped to get the reduced image. Note that in order to maintain uniformity, we need to make sure that all the images are of the same dimensions. As discussed in the later sections, this experiment makes use of two different data-sets, namely MNIST and NUAA. The size of an MNIST image was taken as $28 * 28$ and that of an NUAA data was taken as $64 * 64$.

2. Standardization: Once we have the cropped face, it is essentially to make sure that all the images are comparable. In other words, it has to be normalized/ standardized. The first step in this stage is to convert the existing images to gray-scale. This is non-trivial as in the multi-layered dictionary approach, the statistical features like 2D image intensity and BSIF do not add to contribute to any detail detail. After converting into gray-scale, all the images are normalized such that the intensity values lie in a particular range. Normalization makes sure that there are no differences due to contrast/ glare.

3. Vectorization: As seen in the first step, the images from the two data-sets are of the dimensions $28 * 28$ and $64 * 64$ respectively. These images are then expanded along a single dimension so that they are represented as a $784 * 1$ or $4096 * 1$ vector. In other words, if the initial image is of the dimension $m * n$, then it is flattened to get a dimension of $mn * 1$. Here, m and n are the height and width of the images. These three stages form a part of the preprocessing module. Preprocessing is carried out for all the training and testing data. If the number of samples in the training data-set is N, then the size of the entire training data-set would be (M, mn). All the further computations are done on this 2D matrix.

4. Dictionary Learning: After pre-processing the data, we learn muliple dictionaries to best represent the information in the data. This is further elaborated in section 5.2.2.

5. Classification: The sparse representation of the input data-set is then used to learn a classifier. The performance have been tested on KNN and SVM classifiers. Further details are provided in the later sections.

5 Experimental Setup

5.1 Data-sets

We evaluated our multi-layered dictionary learning model on two data-sets.

5.1.1 NUAA Photograph Imposter Database

This database [4] contains images of real faces and corresponding photographs. The training set has a total of 3,491 images consisting of 1,743 face images and 1,748 images of face photographs.

The testing set has a total of 9,123 images with 3,362 face images and 5,761 images of face photographs. All images are converted to 64 x 64 pixels normalized gray-scale images.

5.1.2 MNIST

The MNIST Data-set [3] contains 70,000 images of handwritten digits (zero through nine), divided into a 60,000-image training set and a 10,000-image testing set. Due to computational reasons, we used a 40,000-image training subset and a 8,000-image testing subset for evaluating our implementations. The images themselves are 28x28 pixel images.

Data-set	Training Size	Testing Size
NUAA	3,491	9,123
MNIST	40,000	8,000

5.2 Methodology

5.2.1 Dictionary Learning

In the vanilla formulation of dictionary learning we learn a dictionary D and sparse representation Z for representing the data X. The columns of D are called *atoms*. We can get back the data from the learnt dictionary along with the sparse representation. Dictionary learning solves the optimization problem in equation (1).

$$\min_{D,Z} \quad ||X - DZ||_F^2 \tag{1}$$

The method of optimal directions is used to solve (1). This is an alternating minimization algorithm where the representation Z is updated by fixing dictionary D and then solve for D assuming Z is fixed.

$$Z_k \leftarrow \min_Z ||X - D_{k-1}Z||_F^2 \tag{2}$$

$$D_k \leftarrow \min_D ||X - DZ_k||_F^2 \tag{3}$$

5.2.2 Multi-layered Dictionary Learning

Let D_1 be the dictionary at the first level, X be the input data and W be the sparse representation.

$$X = D_1 Z \tag{4}$$

This is similar to shallow one-level dictionary learning problem.

Extending the shallow set-up to the second layer, we get

$$X = D_1 \varphi(D_2 Z) \tag{5}$$

Here, φ is the activation function.

With N layers, the deep dictionary learning can be written as:

$$X = D_1 \varphi(D_2 \varphi(\ldots \varphi(D_N Z))) \tag{6}$$

The full optimization can be written as:

$$\min_{D_1, D_2, \ldots, D_N, Z} \quad ||X - D_1\varphi(D_2\varphi(\ldots\varphi(D_N Z)))||_F^2 \tag{7}$$

For the first layer, Z_1 is expressed as $\varphi(D_2\varphi(\ldots\varphi(D_N Z)))$, so that the problem is formulated as:

$$\min_{D_1, Z_1} \quad ||X - D_1 Z_1||_F^2 \tag{8}$$

After the coefficients for the first layer are learned, the learning of second layer can be formulated as:

$$\varphi^{-1}(Z_1) = D_2 Z_2, \text{ where } Z_2 = \varphi(D_3 \ldots \varphi(D_N Z)) \tag{9}$$

and the optimization is:

$$\min_{D_2, Z_2} \quad ||\varphi^{-1}(Z_1) - D_2 Z_2||_F^2 \tag{10}$$

The dictionary learning problem is generally accompanied with sparsity imposing constraints:

$$\min_{D_1, D_2, \ldots, D_N, Z} \quad ||X - D_1\varphi(D_2\varphi(\ldots\varphi(D_N Z)))||_F^2 + \lambda||Z||_1 \tag{11}$$

Thus, for the final layer, we impose sparsity and the optimization problem becomes:

$$\min_{D_N, Z} \quad ||\varphi^{-1}(Z_{N-1}) - D_N Z||_F^2 + \lambda||Z||_1 \tag{12}$$

Like all the above optimization problems, this also can be solved using alternate minimization:

$$\min_{Z} \quad Z_k \leftarrow ||\varphi^{-1}(Z_{N-1}) - D_N Z||_F^2 + \lambda||Z||_1 \tag{13}$$

$$\min_{Z} \quad D_k \leftarrow ||\varphi^{-1}(Z_{N-1}) - D_N Z||_F^2 \tag{14}$$

5.2.3 Effect of various activation functions

The proposed algorithm can use any differentiable activation function. While [1] and [2] use only linear activation functions, we tried the algorithm with non-linear activation functions as well. Since we use inverse of the activation function at each layer, we project the sparse representation at each layer to the domain of the inverse activation function, e.g. [-1, +1] for $arctanh$. Unfortunately, the accuracies obtained via non-linear activation functions is very low in comparison to the accuracies obtained with linear activation functions.

Table 1: Dimensionality Reduction using Dictionary Learning

Dataset	**Feature Size (1 Layer)**	**Feature Size (2 Layer)**	**Feature Size (3 Layer)**
NUAA	512	1024 - 512	1024 - 512 - 256
MNIST	100	200 - 100	400 - 200 - 100

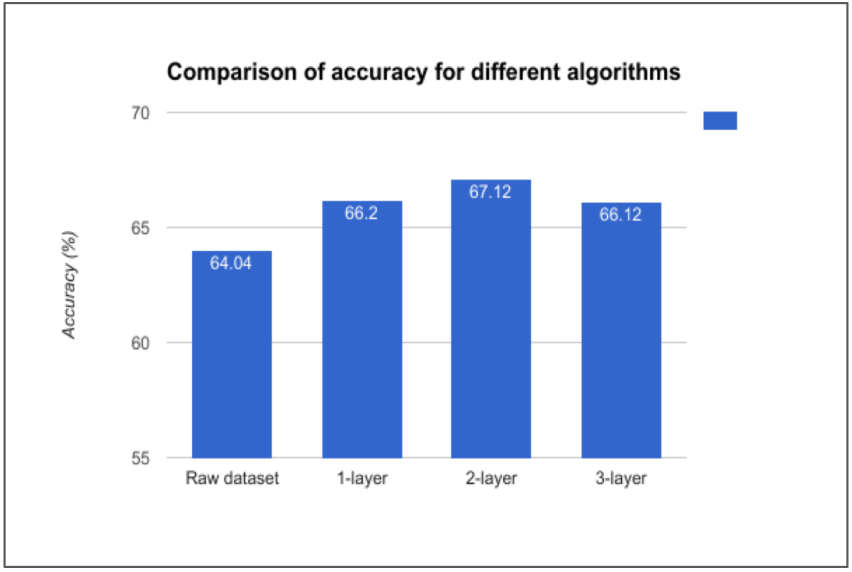
6 Results

6.1 NUAA

The images from NUAA dataset are either images of photographs or images of real faces. We tried classifying them using just the raw images as input and compared it with the different sparse representations learned using different number of dictionary layers.

Classifying just the raw images using k-NN gave an accuracy of 64.04. Though the accuracy is less its still better than guessing, note that its very difficult to classify the two images since they look exactly the same. We use multi-layered dictionaries with the aim that different number of

Figure 2: Classifier accuracy with different sparse representations of NUAA data.

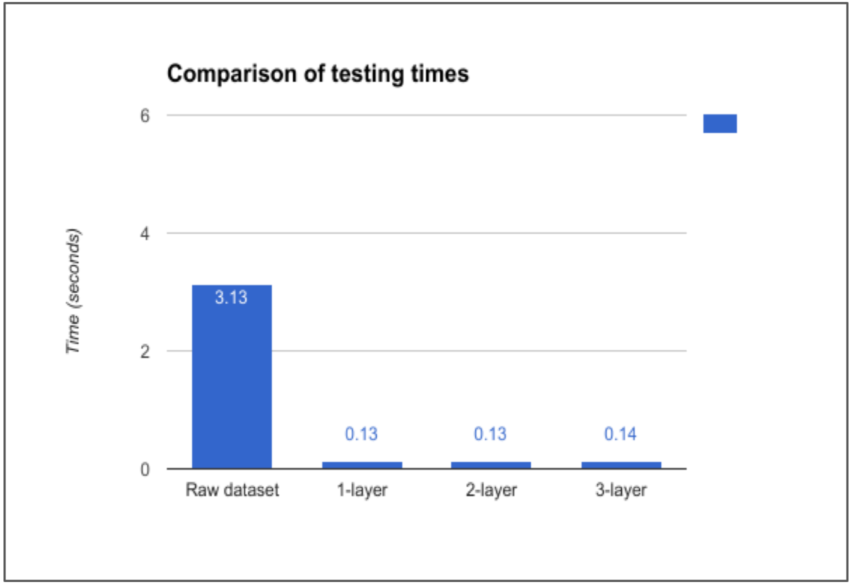


components in the dictionaries will capture information about the image in different dimensions. The learned sparse representations make the classification faster due to the sparsity in the data.

As shown in figure 2, a sparse representation obtained using one-layered dictionary with 512 components gave an accuracy of 66.2%. A two-layered dictionary with 1024 components in the first layer and 512 components in the second layer gave an accuracy of 67.12%. The three-layered dictionary with 1024, 512 and 256 components in its first, second and third layer gave an accuracy of 66.12%.

As seen in figure 3, using the sparse representations improves the classifier testing times significantly along with improving accuracy. Using sparse representations from layered dictionaries is more than 20 times faster than using raw images.

Figure 3: Classifier accuracy for different sparse representations of NUAA data.

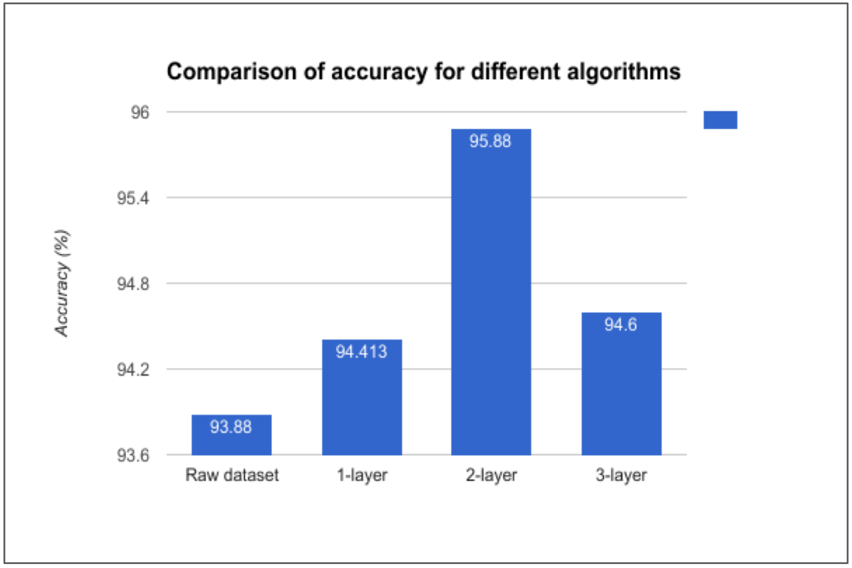


6.2 MNIST

We evaluated the classification accuracy on MNIST dataset, first we just used the raw images itself for classification to obtain a base line then we tried with different sparse representations of the data. We evaluated the accuracy of classifiers using sparse representations using one, two and three layered dictionary learning.

As seen in figure 4 On classification using just the raw images, k-NN gave an accuracy of 93.58%. The one layered dictionary learning is same as the vanilla dictionary learning. We first learn a basis with 100 components and the sparse representation of the data . This sparse representation is used as input to the classifier. For this, k-NN classifier gave an accuracy of 94.41%. In the 2-layered case, for the first layer we learnt a dictionary (D_1) with 200 components and t second layer dictionary (D_2) with 100 components. This gave an accuracy of 95.88%. In the three layered case, we had 400 components in the first layer, 200 components in the second layer and 100 components in the third layer. This gave a classification accuracy of 94.6%.

Figure 4: Classifier accuracy for different sparse representations of MNIST data.



The testing times of the classifier using different representations is as seen in figure 5. Using the raw images it tool 7.15 seconds to classify 8000 images, but with the sparse representations it took 1.29, 0.64, and 0.67 seconds respectively with one layered, two layered and three layered dictionaries. The sparse representations obtained from layered dictionaries were not only faster to train and test by about 10 times compared to raw images, but were also more accurate by about 2%. With more optimizations we expect to get even better results.

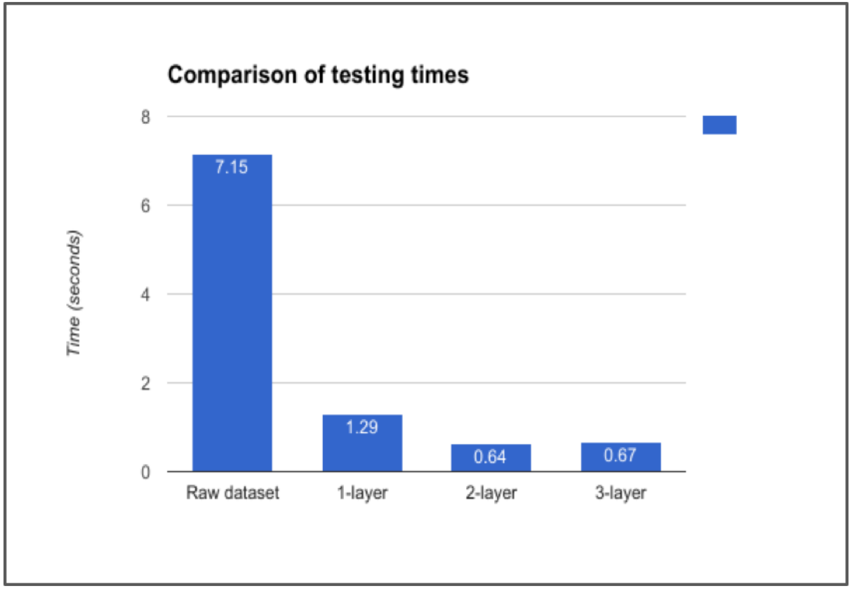
7 Conclusion and Future work

We showed that using multi-layered dictionary learning not only improves the accuracy of classifier by capturing more information but also reduces the classifier training and testing times.

We also note that learning different levels of dictionaries along with the coefficients is not the same as learning a single (collapsed) dictionary and its corresponding features (even with linear activation functions). This is because the single-level problem (single level) is a bi-linear problem and 2-level problem is a tri-linear problem.

A challenge with multi-layered dictionary learning is the guarantee of convergence. Studies have proven the convergence guarantees for single level dictionary learning. These proofs are very hard to replicate for multiple layers. Using greedy approach for layer-wise learning ensures the convergence of each layer because each layer then becomes a shallow dictionary learning problem.

Figure 5: Classifier testing times for different representations of the MNIST data.



Another challenge is that the number of parameters required to be solved increases when multiple layers of dictionaries are learned simultaneously. With limited training data, this could lead to overfitting. We used regularization at final layer to control sparsity and prevent overfitting.

In the future, we'll be using images in 2D format instead of a vector. The data will then become multi-dimensional which calls for multi-dimensional dictionary learning. Also, we'll try using other kind of sparsity measures and comparing performance and also try improving performance for non-linear activation functions.

References

[1] Ishan Manjani, Snigdha Tariyal, "Detecting Silicone Mask based Presentation Attack via Deep Dictionary Learning", *IEEE Transactions on Information Forensics and Security, March 2017, Vol PP, Issue 99*

[2] Snigdha Tariyal, Angshul Majumdar, "Deep Dictionary Learning", *IEEE May 2011*

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition.", *Proceedings of the IEEE, 86(11):2278-2324, November 1998.*

[4] Tan, Xiaoyang, Yi Li, Jun Liu, and Lin Jiang. "Face liveness detection from a single image with sparse low rank bilinear discriminative model." *Computer Vision ECCV 2010 (2010) 504-517. Harvard*

[5] J. Li, Y. Wang, T. Tan, and A. K. Jain, "Live face detection based on the analysis of fourier spectra," *Proc. SPIE: Biometric Technology for Human Identification, 2004, pp. 296303.*

[6] Di Wen, Hu Han, Anil K. Jain, "Face Spoof Detection with Image Distortion Analysis," *IEEE Transactions on Information Forensics and Security, 2015.*

[7] R Raghavendra, Christoph Busch, "Presentation Attack Detection Algorithm for Face and Iris Biometrics," *Eurasip Proceedings.*

[8] Jun Liu, Ajay Kumar, "Detecting Sensor Level Spoof Attacks Using Joint Encoding of Temporal and Spatial Features," *IEEE 2016*

[9] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE, 2001.*

Additional materials

Code and demo are provided here:
https://github.com/atif93/AMLProject