Is There An Imposter Among Us?
Face Liveness Detection Using Generative Adversarial Networks

A Project Report Submitted in Partial Fulfillment of the Requirements for the Award of the Degree of

Bachelor Of Technology
in
COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Satashree Roy (17-1-5-010), Menakuru Meghana (17-1-5-043), Ashutosh Singh (17-1-5-080), Shuvo Jyoti Sarkar (17-1-5-004)

Under the Supervision of:

Mr. Pantha Kanti Nath
Assistant Professor
Department of Computer Science & Engineering
NIT Silchar

Department of Computer Science & Engineering
National Institute of Technology Silchar
Cachar District, Assam, India (788010)

Department of Computer Science & Engineering
National Institute of Technology Silchar

DECLARATION

Thesis Title: **"*Is There An Imposter Among Us?* Face Liveness Detection Using Generative Adversarial Networks"**

Degree for which the Thesis is submitted: Bachelor of Technology

We certify that this thesis is done with our own personal research and that we have not copied from any sources. We have acknowledged all the resources before preparing our thesis, whether they are books, articles, lecture videos, or any other forms of online or face to face interaction and consults. We also certify that this thesis has not been submitted for assessment prior to any other unit, except where specific permission has been granted from all unit coordinators involved.

Satashree Roy (17-1-5-010)
Menakuru Meghana (17-1-5-043)
Ashutosh Singh (17-1-5-080)
Shuvo Jyoti Sarkar (17-1-5-004)
Wednesday 12th May, 2021

Department of Computer Science & Engineering
National Institute of Technology Silchar

CERTIFICATE

This is to certify that the project titled **"Is there an Impostor Among us? Face Liveness Detection using Generative Adversarial Networks"** is the bona fide work carried out by **Satashree Roy (17-1-5-010), Menakuru Meghana (17-1-5-043), Ashutosh Singh (17-1-5-080) and Shuvo Jyoti Sarkar (17-1-5-004)**, students of BTech CSE of National Institute of Technology, Silchar, India (An Institute of National Importance) during the academic year 2020-2021, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology (Computer Science and Engineering).

The work presented here has not previously formed the basis of the award of any other degree, diploma, fellowship or any other similar title.

Mr. Pantha Kanti Nath
Assistant Professor
Computer Science & Engineering
National Institute of Technology Silchar
Wednesday 12th May, 2021

Acknowledgement

Any accomplishment requires the effort of many people and this work is no different. First of all we would like to acknowledge our families who supported us with love and understanding that led us to this level of success.

Secondly, we would like to express our sincere gratitude to our project guide and mentor, **Mr. Pantha Kanti Nath**, Assistant Professor, Department of Computer Science & Engineering, National Institute of Technology Silchar, for his kind and continuous support for this work of study. This thesis would not have been possible without his encouragement throughout. We would also like to thank **Dr. Samir Kumar Borgohain**, Head of the Department, Department of Computer Science & Engineering, for giving us the opportunity of working on such a project.

Finally, we would like to state our thanks to our friends and batchmates who have always been there for us in times of need.

Every effort has been made to give credit where it is due for the material contained and we express our gratitude to everyone who may contributed to this work, even though anonymously.

Contents

List of Figures

List of Tables

Abstract

Over the past few years, face recognition systems have achieved very high accuracies in addition to being fast and light enough to be implemented in commercial authentication devices. However, in order to be truly adopted as a credible means of biometric authentication it is paramount to be able to distinguish between a genuine live face registered in the system and an attempt to fool the system using spoof-attacks. Face recognition systems can be defeated using impostor attacks such as printed photographs, hyper-realistic animation, 3D masks, etc. Therefore, it is essential to have a liveness detection or, anti-spoofing component in the system that determines if the sample is real or fake. While most conventional methods require extensive feature engineering, our approach would be to use the Adversarial machine learning framework, specifically Generative Adversarial Networks(GANs) as it is intuitive considering that spoof samples are the adversaries that the system must detect and avoid. Our approach is free of the spoof faces, thus being robust and general to different types of spoof, even unknown spoof attacks.
Empirical results show that the proposed model performs significantly better than the baseline model.

Keywords: Authentication, face recognition, spoof attacks, autoencoder, generative adversarial networks

Chapter 1

Introduction

When it comes to Face Recognition Systems (FRS), accuracy and performance are now no longer a concern. The computer vision research community, supported by publicly available huge datasets and powerful computing resources, have successfully produced state-of-the-art FRS with accuracy as high as 99.63% (Google FaceNet [1]). Hence, face biometrics is gaining popularity as a convenient method of identity verification. Capturing face biometric sample is also much more user-friendly. However, FRS have a critical flaw- they are vulnerable to spoofing and can be defeated using various presentation attacks such as printed photographs, pre-recorded video attacks, face masks (with eye and lip cut-outs to bypass systems that rely on eye-blinking and lip-moving), 3D models, deliberate pixel distortions etc. The seminal work "One Pixel Attack for fooling deep neural networks" by [2] *et al.*, shows that the minimum number of pixels needed to be modified to defeat a Deep Neural Network is just 1. This shows the inherent vulnerability of deep learning systems, which has given rise to an exciting framework of machine learning known as "Adversarial Machine Learning" [3]

If a potential impostor can easily access an image of a person's face and present it as their own, the authentication system would lose its reliability. Hence, a comprehensive face biometric system must have two components in order to be complete:
a) Biometric matching (FRS),
b) Liveness detection or anti-spoofing.
Research on liveness or presentation attack detection has increasingly gained momentum and a number of methods have been developed and deployed. Conventional methods usually consist of extensive hand-crafted feature engineering, which aim to derive useful features and pass those into a classification model that learns to differentiate between real and fake images based on these features. These methods are likely to be overfitted to one particular setup which may limit their application in practical scenarios when confronting diverse image or video capturing conditions as well as various kinds of spoof attacks. Deep learning

techniques which can extract rich semantic information, have led to major improvements in performance but are subject to large amount of data. Moreover, DNNs could learn incorrect associations between nuisance factors in the raw data and the target. For example, pose, gender, skin tone etc. are nuisance factors for liveness detection. Such limitations lead to poor generalization and performance in cross-dataset settings.

A fairly recent shift has been towards using adversarial learning and generative adversarial networks (GANs) architecture for presentation attack detection. GANs [4] are an approach to generative modelling that use two sub models, a "generator" that is trained to produce new samples which resemble the original data distribution and, a "discriminator" that learns to distinguish between the real samples and the samples generated by the generator. Both the models are trained together in a minimax game and keep getting better at their task; as such, the generator is able to model the input distribution well and produce new real looking samples.

In our work we exploit this aspect of the GAN architecture in order to model the distribution of live samples and learn the necessary features. We take a convolutional autoencoder as our generator and a binary (real/fake) classifier as our discriminator. The elementary idea is that, since the generator is trained only on the real face images, the autoencoder shall yield higher reconstruction error when given to reconstruct a spoof image. By training the generator only on real samples we make it learn the features of real samples while it is completely unaware of the distribution of spoof samples. In this way we eliminate the need of collecting annotated spoof samples, which is costly, as well as account for all kinds of spoof attacks-possibly even attacks presently unknown. The major contributions of our work will be as follows:

- A GAN based liveness detection system that requires only live samples and is expected to guarantee security against spoof attacks.
- Proposed algorithm will be trained and tested against benchmark datasets.

1.1 Motivation and Objective

As mentioned above vulnerability of FRS models to spoof attacks is an obstacle towards the complete adoption of machine learning based face biometric systems. Selection of the right kind of features that will give the most quality results is a challenge. Existing techniques are specific and may not account for all the different kinds of spoof attacks, and show limited robustness, generalization and poor performance in cross-dataset settings. Moreover, collecting a diverse well annotated dataset of spoof images is a costly and cumbersome task.
We propose to utilize GANs in our approach because it is intuitive for our problem statement, in the sense that live images are real data and spoof images are adversaries that the system

must learn to avoid. Our approach is unsupervised and does not require labelled spoof data; i.e it requires only the real face images. In addition to that, the model can learn a very good representation of the data and hence learn useful inherent features that can model the live samples well. Another reason for exploring GANs towards this end is that it is currently a hot topic of interest in the research community. However, a complex network and unstable training pose as challenges while dealing with GANs.

Chapter 2

Literature Survey

Image based presentation attack detection methods have conventionally involved extensive hand-crafted feature engineering and the extraction of discriminative features which are then analysed to distinguish genuine images from spoof images. Kim *et al*. [5] exploited the difference in illuminance component and texture richness using power spectrum or local binary patterns (LBP). Support Vector Machine (SVM) classifier was used for learning liveness detectors with the feature vectors generated. Wang *et al*. [6] also make us of the features derived from the Fourier spectra of the images. The optical flow field based technique, introduced by Bao *et al*. [7] analyses the differences and properties of optical flow generated from 3D objects and 2D planes. It is assumed that spoof attacks are 2D in nature; a reference field is deduced from the optical field and the difference between the two is used to decide whether a face is real or not. S. Kim *et al*. [8] utilized the variation in pixel values by focusing between two images sequentially taken in different focuses. In case of real faces, the focused regions are found to be clear than the rest due to depth information. In contrast, there is little difference between images taken in different focuses from a printed copy of a face. Some of the works in the literature have also employed features from different colour spaces [9]. These texture based and statistical machine learning techniques are generally specific to one or two types of attacks and it becomes impractical to account for all the different kinds of spoof attacks possible. Moreover, such detection techniques may be highly influenced by external conditions like illumination, resolution, shooting angles, etc.

Deep learning has brought new levels of performance in a plethora of application domains. Convolutional neural networks (CNNs), capable of learning inherent discriminative features, have been successfully employed for liveness detection [10]. A novel framework that considered both spatial and temporal information and a 3D CNN for learning generalized feature representation, giving good cross-dataset results, was proposed in [11]. Rehman *et al*. [12]

propose and evaluate the adaptive fusion of features learned by convolutional layers from real face images and deep CNN generated face images. The original image and the autoencoder generated image are passed through the convolutional layers with shared weights followed by an adaptive convolutional fusion layer that adaptively weight the incoming feature maps and passed it to later layers of CNN classifier. As mentioned before, DNNs may learn incorrect associations between irrelevant features and final target, which limits its robustness. In the work [13] the authors propose "RoPAD", a novel DNN model that robustly classifies real and fake images by being invariant to nuisance factors in images. The invariance is achieved by adopting the unsupervised adversarial invariance framework that induces implicit feature selection and invariance to visual distractors without requiring nuisance annotations. Spoofing methods are also improving day by day and the fake images are getting closer to the real ones. Li et al. [14] show that the real and fake face images highly overlap in the original colour space. Inspired by GAN, they propose an end-to-end deep learning network to generate a new colour like space where the real and fake samples are as separate as possible. A feature extractor then obtains features from this new learnt colour space, which are then fed into an SVM classifier. We tabulated the data of various face liveness detection methods developed so far in Table 2

Most of the existing approaches use a two-class formulation where they obtain some useful features and separate the negative (fake) from the positive (genuine) samples. This requires annotated data of both classes collecting which is costly and time consuming. The proposed methodology is a GAN based model that will be trained on only genuine face images.

Table 2.1: Characteristics of Different face liveness detection methods

Methods	**Analysis Type**	**Strategy**
LBP	Texture analysis	Micro-texture analysis via LBP with SVM as a classifier
DoG-SL	Texture analysis	Applying an adaptive histogram equalisation to the images
Color-texture	Texture analysis	Computing a half of Face with another half that is divided in two ways: horizontally and vertically
Optical flow field	Motion analysis	Analyzing the optical flow field to detect real face
Structure tensor	Motion analysis	Face motion estimation based on the structure tensor and a few frames
Spatial-temporal Domain	Motion, Texture analysis	A two-stream structure spatial, temporal)
Patch-based CNN	Texture, cue analysis	Extracting the local features and holistic depth maps from the face images
VGG	Texture analysis	Extracting the deep partial features from the convolutional neural network CNN
Liveness optical flow	Motion analysis	Applying the Support Vector Machine to distinguish between the motion information of real faces and photographs
Auxiliary	Texture, cues analysis	Fusing the estimated depth and rPPG to distinguish live vs spoof faces
De-Spoof	Texture analysis	A CNN architecture with proper constraints and supervisions

Chapter 3

Proposed Methodology

In this section we present our GAN based technique introduced earlier. We begin by describing our basic idea, followed by an overview of the network architecture and our expectations.

The underlying idea behind our project is that, since the generator of our GAN is trained only on genuine face images, the learned model is therefore inferior in depicting the characteristics of spoof face data. This difference can be exploited to distinguish between real face and spoof face. We take a convolutional autoencoder (CAE) [15] as the generator. Autoencoders (AE) are data-specific in nature; an AE trained to regenerate a particular type of data gives very high reconstruction error when given to generate some other kind of data. We leverage this limitation of AE to our advantage. A normal binary (real/fake) classifier is taken as the discriminator.

3.1 Dataset and Preprocessing

Liveness detection has gained a new traction in recent years and a number of well curated datasets and benchmarks have been created for the same. The ROSE-Youtu Face Liveness Detection Dataset [16] consists of 4225 videos with 25 subjects in total (3350 videos with 20 subjects publicly available). The CASIA-SURF [17] dataset contains 21000 videos in total of 1000 subjects with each sample having 3 modalities (RGB, Depth and IR). The CASIA FaceV5 has 2500 color facial images of 500 subjects. The PRINT-ATTACK [18] database contains 200 videos of real-accesses and 200 videos of spoof attempts using printed photographs of 50 different identities. The authors also provide a baseline motion-based algorithm that detects correlations between the person's head movements and the scene context. The REPLAY-ATTACK [19] provides a more robust testing as it contains 1300 video clips

of photo and video attack attempts under different lighting conditions. In our work we use the easily available ROSE-Youtu dataset. It is to be noted that most of the available datasets have faces of a particular ethnicity and hence the learned model will show poor generalization. Training the model with a diverse dataset is encouraged.

3.2 Generative Adversarial Networks

GAN is a class of generative models designed by Goodfellow *et al.* [4] that simultaneously trains two networks: a generator model G that tries to capture the input distribution and a discriminator model D that estimates the probability that a sample is from the training data rather than generated by G. GAN takes a game-theoretic approach, with both the networks competing against each other optimally; the aim of G is to successfully fool D by generating real looking samples while the aim of D is to successfully distinguish between real and generated samples. G and D play the minimax game with the following objective:

$$J(G,D) = \min_G \max_D \left\{ \mathbb{E}_{x \sim P_{data}}[\log(D(x))] + \mathbb{E}_{z \sim P_z}[\log(1 - D(G(x)))] \right\} \tag{3.1}$$

$D(x)$ is the Discriminator output for the real data x, and $D(G(z))$ is the Discriminator output for the generated fake data $G(z)$. P_{data} is the distribution of the real data whereas $P(z)$ is the distribution of the data generated by the Generator. The Discriminator D has to maximize the objective such that $D(x)$ is close to 1 (real) and $D(G(z))$ is close to 0 (fake); Generator G has to minimize the objective such that $D(G(z))$ is close to 1 (fooled into thinking generated $G(z)$ is real).

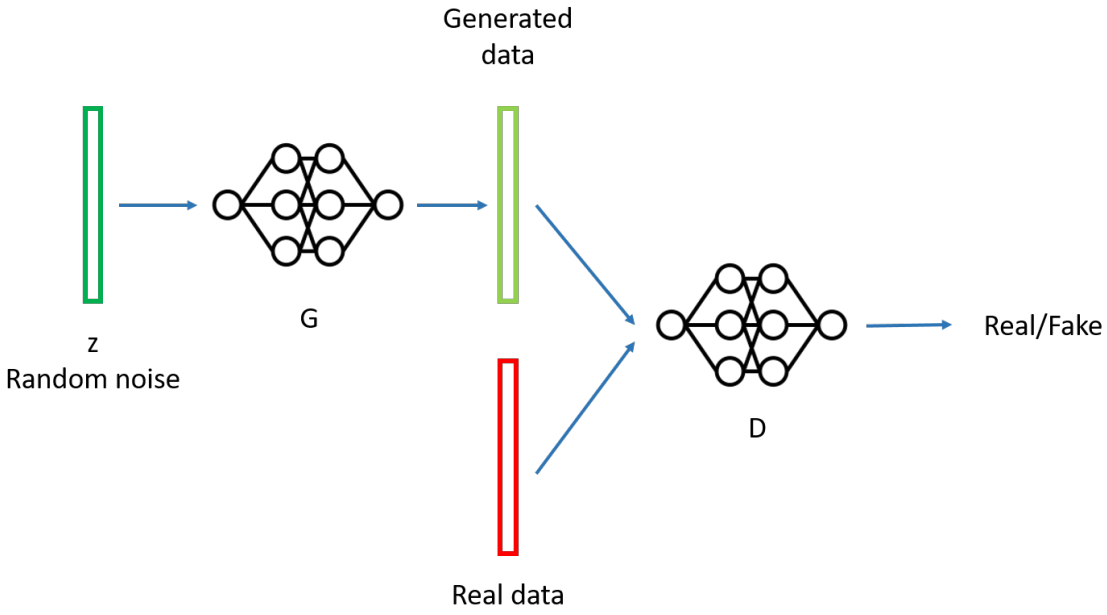

Figure 3.1: Basic GAN architecture

The Discriminator and the Generator are trained alternatively as follows:

1. **Gradient ascent** on D.

$$\max_D \left\{ \mathbb{E}_{x \sim P_{data}} [\log(D(x))] + \mathbb{E}_{z \sim P_z} [\log(1 - D(G(z)))] \right\} \tag{3.2}$$

2. **Gradient descent** on G.

$$\min_G \mathbb{E}_{z \sim P_z} [\log(1 - D(G(z)))] \tag{3.3}$$

In practice, optimizing this Generator objective does not give good results as the gradient signal is dominated by the region where the sample is already good and the problem of vanishing gradient arises. The generator loss is therefore modified as follows:

$$\max_G \mathbb{E}_{z \sim P_z} [\log(D(G(z)))] \tag{3.4}$$

for *num_iters* **do**
  for *ksteps* **do**
    - sample minibatch of m noise samples $z^1, z^2, ..., z^m$ from noise prior $P(z)$
    - sample minibatch of m real examples from P_{data}
    - update discriminator by ascending its stochastic gradient:

$$\frac{1}{m} \sum_{i=1}^{m} [\log(D(x^i)) + \log(1 - D(G(z^i)))]$$

  end

  - sample minibatch of m noise samples $z^1, z^2, ..., z^m$ from noise prior $P(z)$
  - update the generator by ascending its stochastic gradient:

$$\frac{1}{m} \sum_{i=1}^{m} [\log(D(G(z^i)))]$$

end

Algorithm 1: GAN training algorithm

3.3 Convolutional Autoencoder (CAE)

As mentioned before, we take a CAE as our generator G. An autoencoder is a neural network architecture capable of discovering structure within data in order to develop compressed representation of the input by imposing a "bottleneck" in the network. This compressed representation is expected to characterize meaningful attributes of the original data input. The bottleneck constrains the amount of information that can traverse the full network, forcing a learned compression of the input. The network takes unlabelled data x and frames it as a supervised learning problem with the objective of outputting $\hat{x}$ which is a reconstruction of original input x. The network is trained by minimizing the reconstruction error:

$$L(x,\hat{x}) = ||x - \hat{x}|| \tag{3.5}$$

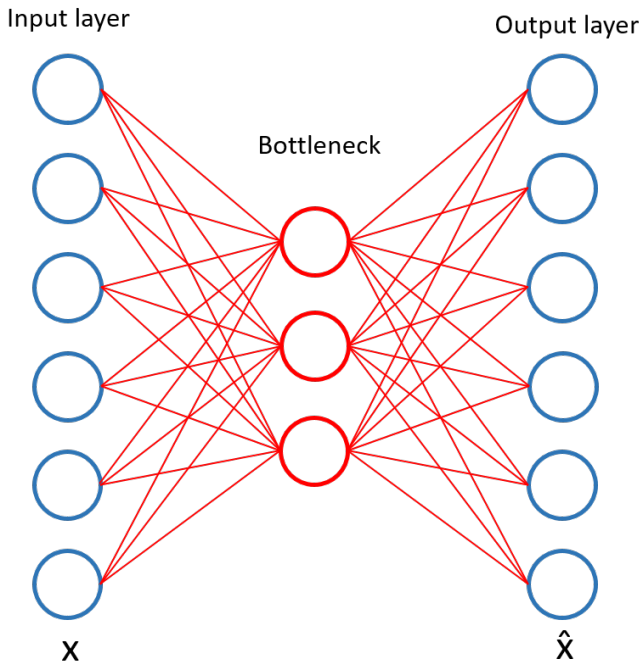

Figure 3.2: Autoencoder architecture

3.4 Network Architecture

The figure below gives an overview of the network architecture. We will train our model with two kinds of input- a) Direct real face images (i.e. considering only the appearance information) and, b) Optical flow maps (i.e. considering only the motion information). The input will be fed to the CAE for reconstruction. The reconstructed input (i.e. output of the CAE) and the original input will be passed to the binary classifier to distinguish between real

and generated data. The CAE and the classifier will be trained simultaneously in accordance with the GAN framework. After the CAE is trained, the average reconstruction error will be noted. For spoof examples, the classifier will be removed from the model and the spoof image will be fed to the CAE for reconstruction. The expectation is that, the CAE which has learned optimal filters for the reconstruction of real faces, will give high reconstruction error (above the aforementioned threshold) for the spoof faces. In this way it can be determined if the input sample is spoof or real.

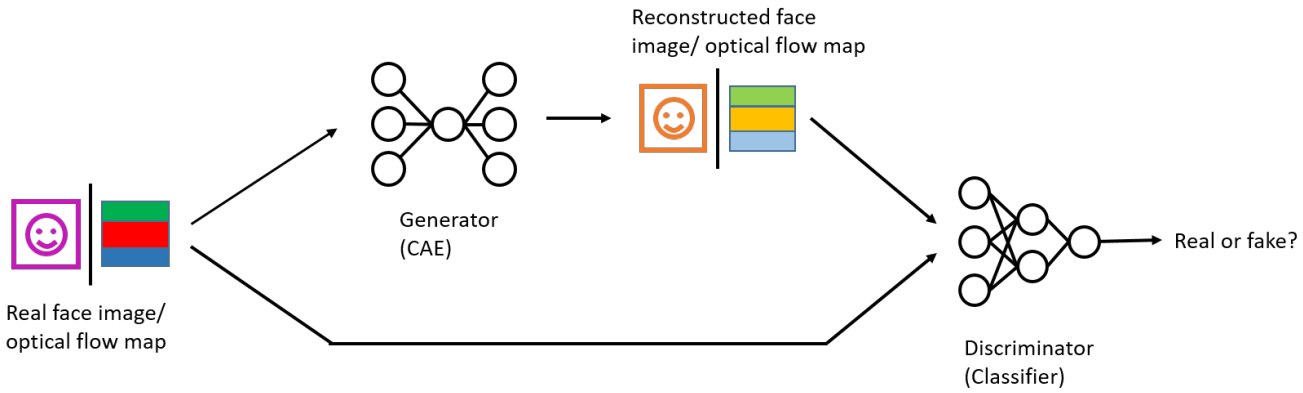

Figure 3.3: Proposed network architecture

3.5 Overall Loss

To train our model we take loss function defined as a combination of eq(3.1) and eq(3.5)

$$L_{tot} = J(G,D) + L(x,G(z)) \tag{3.6}$$

Chapter 4

Experimentation and Results

In this section we provide the experimentation details.

4.1 Experimental set-up and Data-preprocessing

The baseline model experiments have been performed on Lenovo Ideapad 510 laptop i5 processor with 8 GB RAM using Python and Tensor-flow/Keras deep learning framework. For the final model we use Nvidia GeForce 940MX GPU with 4GB dedicated RAM. As mentioned before, we make use of the ROSE-Youtu dataset for Training and Experimentation purposes.
Frames were extracted, with a step-size of 80, from the 897 genuine face videos using the python package named *detecto*. A dataset of 4034 real face images was hence created. A good face anti-spoofing system shall be robust to different types of spoof attacks. Hence we evaluate our Baseline and proposed model considering varied spoof attacks dataset. With the available Device, Mask Print attack videos, frames were extracted. Altogether 4034 genuine face images and 11,717 spoof images were collected. Among the many spoof images extracted, we utilised 800 of the randomly chosen images for the testing purpose. To standardise as well as to reduce the number of units in deep learning model (to reduce the time complexity), all the images(frames) were resized to a dimension 360×640.

4.2 Baseline Model

The baseline model is simply the convolutional autoencoder, i.e the generator part of the proposed model.

4.2.1 Model Architecture

We create a 12-layer deep convolutional autoencoder comprising of 4963 trainable parameters in total. A convolution operation is the simple application of a filter to an input that results in an activation. Repeated application of the same filter to an input results in a map of activations called a feature map, indicating the locations and strength of a detected feature in an input, such as an image. Convolutional neural nets learn multiple features in parallel for a given input and have the powerful advantage of parameter sharing, i.e sharing of weights by all neurons in a particular feature map which reduces the total number of learnable parameters. The function of pooling layers is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. The most common pooling layer used is the Max Pooling layer. The Upsampling and Transpose convolutional layers are applied to increase the dimensions and perform inverse convolution, starting from the CODE layer (refer to figure 3.2) up to the output layer. The Max pooling and the Upsampling layers have no weights. For an input with dimension $n \times n \times c$, applying d number of filters of size f, padding p and with a stride s results in an output dimension of size $(n+2p-f)/s+1, (n+2p-f)/s+1, d$.

The first Conv layer applies 16 filters with size (3, 3) to the input of dimension (640, 360, 3), resulting in output of dimension (640, 360, 16). It is to be noted that dimension is preserved as "same" padding is used. The MP layer with size (2, 2) downsamples it to dimension (320, 180, 16). This is followed by 8 filter Conv layer, (2, 2) MP layer and again 8 filter Conv layer. This forms the encoder side of the CAE. The CODE layer is a (2, 2) MP layer. For the decoder side, the CODE layer output is upsampled by applying a 8 filter transpose convolutional of filter size (3, 3), followed by a (2, 2) US layer. Another unit of the same ConvTrans and US layer is applied. A 16 filter ConvTrans followed by (2, 2) US layer is applied next. Finally to get the output dimension of (640, 360, 3) a 3 filter Conv layer is applied. All the convolutional and pooling layers have stride=1, padding="same" and layers having weights apply the Relu activation function on the units.

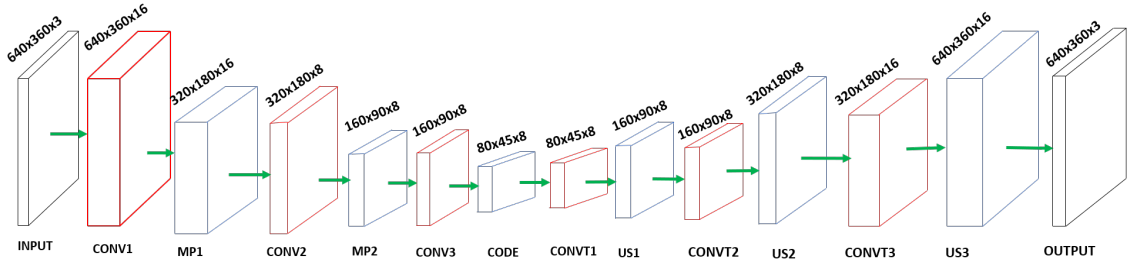

Figure 4.1: Network architecture of the baseline model (CAE)
CONV: Convolutional layer, MP: Max pooling layer, US: Upsampling layer, CONVT: Transpose convolutional layer

4.2.2 Model Training

The model is trained with real face images (i.e. only the appearance information has been considered as of now). The dataset is split into the train set containing 3200 real images and test set containing 834 real images. The Keras *generator class* has been used to generate the input data on the fly so as to avoid loading the entire dataset into the memory at the same time, and loading it batchwise instead. The input is fed into the CAE for reconstruction. The model is trained for 5 epochs with a batchsize of 32, and produces a validation loss of 0.0035.

```
Epoch 1/5
100/100 [==============================] - 1236s 12s/step - loss: 0.0397 - val_loss: 0.0089
Epoch 2/5
100/100 [==============================] - 1299s 13s/step - loss: 0.0073 - val_loss: 0.0069
Epoch 3/5
100/100 [==============================] - 1287s 13s/step - loss: 0.0063 - val_loss: 0.0043
Epoch 4/5
100/100 [==============================] - 1267s 13s/step - loss: 0.0059 - val_loss: 0.0055
Epoch 5/5
100/100 [==============================] - 1247s 12s/step - loss: 0.0046 - val_loss: 0.0035
```

Figure 4.2: Training the baseline model

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
INPUT (InputLayer)           (None, 640, 360, 3)       0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 640, 360, 16)      448
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 320, 180, 16)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 320, 180, 8)       1160
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 160, 90, 8)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 160, 90, 8)        584
_________________________________________________________________
CODE (MaxPooling2D)          (None, 80, 45, 8)         0
_________________________________________________________________
conv2d_transpose_1 (Conv2DTr (None, 80, 45, 8)         584
_________________________________________________________________
up_sampling2d_1 (UpSampling2 (None, 160, 90, 8)        0
_________________________________________________________________
conv2d_transpose_2 (Conv2DTr (None, 160, 90, 8)        584
_________________________________________________________________
up_sampling2d_2 (UpSampling2 (None, 320, 180, 8)       0
_________________________________________________________________
conv2d_transpose_3 (Conv2DTr (None, 320, 180, 16)      1168
_________________________________________________________________
up_sampling2d_3 (UpSampling2 (None, 640, 360, 16)      0
_________________________________________________________________
OUTPUT (Conv2D)              (None, 640, 360, 3)       435
=================================================================
Total params: 4,963
Trainable params: 4,963
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.3: Baseline model summary

4.2.3 Results

For evaluating our Baseline model, we chose, 134 out of 834 real images to calibrate the threshold to distinguish between real and real-reconstructed faces. These images are fed to the CAE for reconstruction and the average difference (d) between the real and real-reconstructed images is calculated to arrive at the threshold t.
We calculated the d for the remaining (700) real and real-reconstructed images. The ones which give a value of d less than the threshold t are the ones that are correctly classified.
We do the same for the chosen 800 spoof images. We reconstruct the spoof images, and we found the d for spoof and spoof-reconstructed images. Those giving d higher than t are correctly classified.
Out of the 700 real faces, we observed that 635 were rightly classified $(d < t)$ by the baseline model and out of the 800 spoof images, 590 were rightly classified to be fake.
We repeated this process by considering different number of real images for calibrating threshold t. The results are shown in the Table 4.2.3. Accuracy in each scenario is calculated and mentioned in the tabular data. It the total number of rightly classified real & spoof-reconstructed images upon the total test data (real + spoof faces).

Table 4.1: Performance of baseline model

No.of real images for calibrating threshold	**Correctly classified real images**	**Correctly classified spoof images**	**Accuracy**
100	87.33%	66.5%	76.47%
134	90.71%	73.75%	81.67%
150	83.91%	76.25%	79.78%
200	76.22%	77.63%	77.06%

Here we outline few examples of real vs. real-reconstructed images, and spoof vs. spoof-reconstructed images.

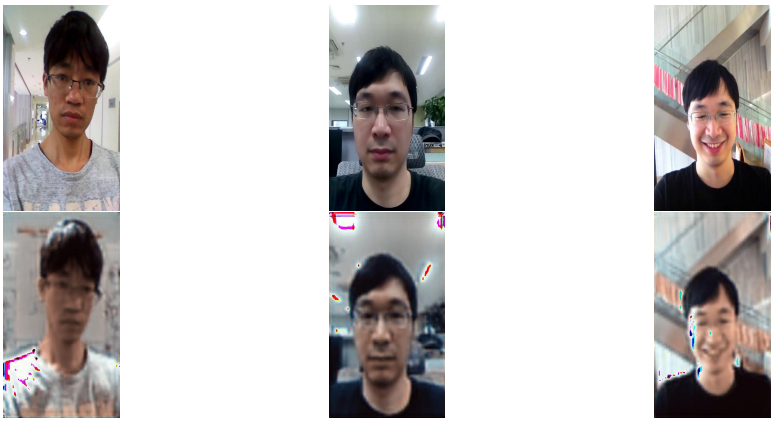

Figure 4.4: Top row shows the real images and the bottom row shows the corresponding reconstructed images by CAE

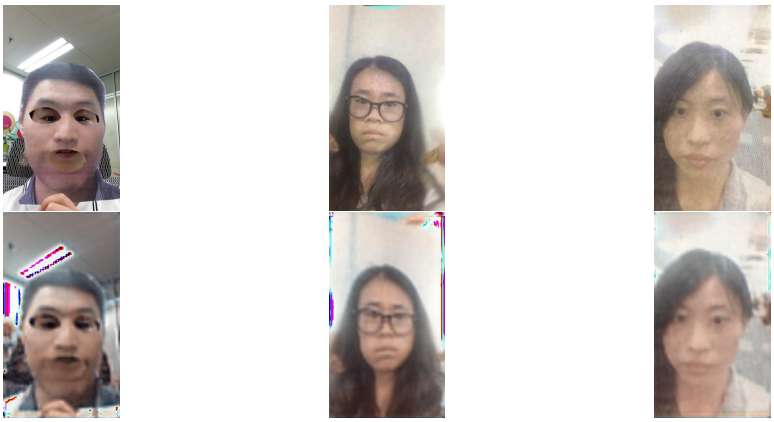

Figure 4.5: Top row shows the spoof images and the bottom row shows the corresponding reconstructed images by CAE

4.3 Our Model

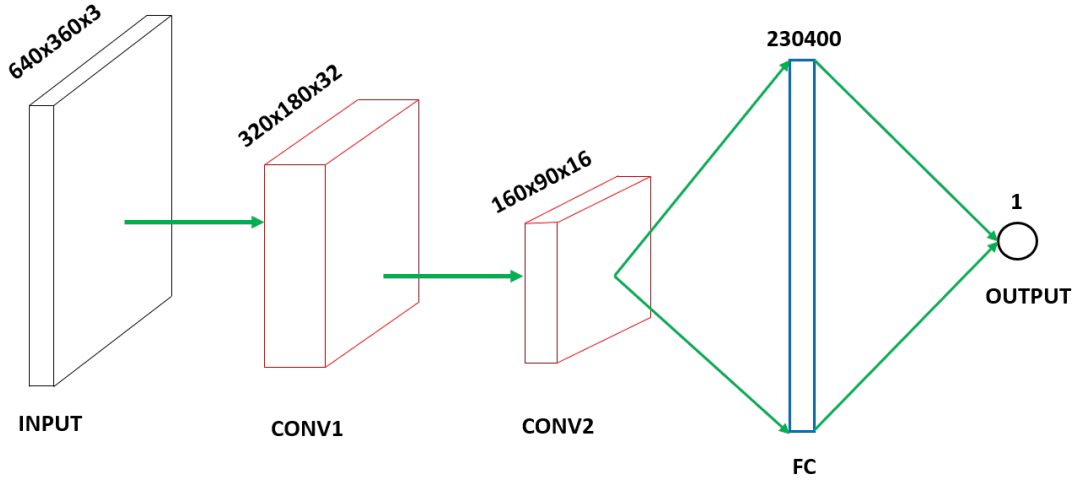

Figure 4.6: Discriminator architecture
CONV: Convolutional layer, FC: Fully connected layer

As explained in section 3.4, the Generator part of the model is a convolutional autoencoder. To this end, we use the same CAE that we had taken as our baseline model. Please refer to figure 4.1 for the architecture of the generator of our GAN. For the Discriminator we use a simple convolutional neural network image classifier with a single output node that produces the probability of the image of being real vs. fake. The first Conv layer applies 32 filters with size (5, 5) in strides of (2, 2) to the input of dimension (640, 360, 3), to produce output of dimension (320, 180, 32). The network units are dropped randomly with a drop

rate of 0.3.

The next Conv layer applies 16 filters of size (5, 5) in strides of (2, 2) to produce output of dimension (160, 90, 16). This layer is then flattened to give a vector of size of 230400 which is densely connected to a single unit in the output layer. Thus the total number of trainable parameters is 245,649. We have used Binary Crossentropy loss for both the generator and the discriminator and have trained the same using Adam optimizer with learning rate = 0.0001.
We have trained the GAN for 50 epochs which took around 5 hours on our system.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 320, 180, 32)      2432
_________________________________________________________________
leaky_re_lu_5 (LeakyReLU)    (None, 320, 180, 32)      0
_________________________________________________________________
dropout (Dropout)            (None, 320, 180, 32)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 160, 90, 16)       12816
_________________________________________________________________
leaky_re_lu_6 (LeakyReLU)    (None, 160, 90, 16)       0
_________________________________________________________________
dropout_1 (Dropout)          (None, 160, 90, 16)       0
_________________________________________________________________
flatten (Flatten)            (None, 230400)            0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 230401
=================================================================
Total params: 245,649
Trainable params: 245,649
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.7: Discriminator model summary

4.3.1 Results

We evaluated our Proposed GAN model, similar to how we have done for our Baseline model (CAE). We considered 134 out of 834 real images to calibrate the threshold to distinguish between real and real-reconstructed faces. These images are fed to the CAE for reconstruction and the average difference (d) between the real and real-reconstructed images is calculated to arrive at the threshold t.
We calculated the d for the remaining(700) real and real-reconstructed images. The ones which give a value of d less than the threshold t are the ones that are correctly classified.
We do the same for the chosen 800 spoof images. We reconstruct the spoof images, and we found the d for spoof and spoof-reconstructed images. Those giving d higher than t are correctly classified.
Out of the 700 real faces, we observed that 664 were rightly classified $(d < t)$ by our model and out of the 800 spoof images, 671 were rightly classified to be fake.
We repeated this process by considering different number of real images for calibrating threshold t. The results are shown in the Table 4.3.1. Accuracy in each scenario is calculated and mentioned in the tabular data. It is the total number of correctly classified real & spoof-reconstructed images upon the total test data (real + spoof faces).

Table 4.2: Performance of our final model(GAN architecture)

No.of real images for calibrating threshold	**Correctly classified real images**	**Correctly classified spoof images**	**Accuracy**
100	94.14%	83.5%	88.59%
134	94.85%	83.88%	89.01%
150	92.69%	82.87%	87.39%
200	93.82%	84.03%	87.93%

Figure 4.8: Top row shows the real images and the bottom row shows the corresponding reconstructed images by GAN

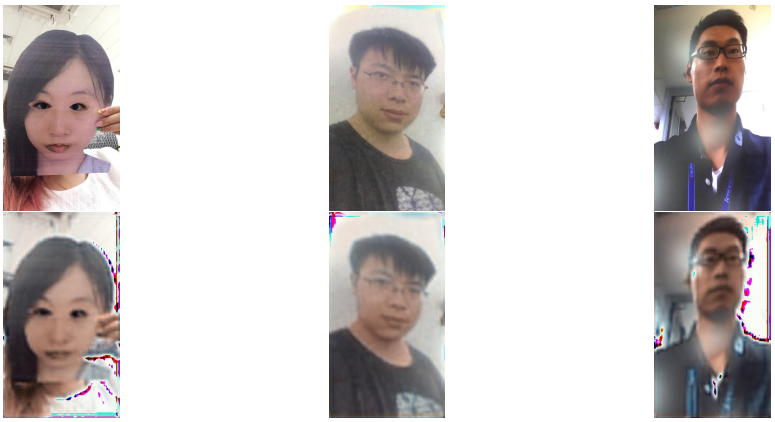

Figure 4.9: Top row shows the spoof images and the bottom row shows the corresponding reconstructed images by GAN

Chapter 5

Conclusion and Discussion

In this work, we presented an approach for the face liveness detection problem to detect various spoof attacks using the adversarial learning framework.

Given an input face image, the trained model can automatically detect whether it is a live or a spoof face. Currently available face anti-spoofing techniques utilise both real and spoof face data for training purpose. By contrast, our approach does not need spoof data for training, and is thus semi-supervised and robust to different types of spoof attacks, of those perhaps unknown. We created the Baseline model with a simple convolutional autoencoder and have taken the reconstruction error between the given image and the reconstructed image as a measure to identify the class of the image. Specifically, low reconstruction error implies that the given image is a real face and high reconstruction error implies that the given image is a spoof face. We constructed our proposed model, i.e., Generator-Discriminator approach, and trained both the models with only the genuine face images. The models have been evaluated with test dataset of real as well as spoof image, which includes variety of spoof attacks. The effect of adding the discriminator to the CAE have been analysed, in order to demonstrate the effectiveness of adversarial learning. The comparative study shows that our proposed method achieves significantly better performance than the baseline.

We believe that Generative Adversarial Networks is one of the most exciting advancements in Machine Learning in recent years, and Face Liveness Detection as one of the applications of GANs adds a new perspective over the present techniques available to detect spoof attacks. The code is available here [1] [2]

[1] Code for CAE
[2] Code for GAN

Chapter 6

Future Scope

As highlighted in section 3.1 the dataset used consists of faces from a particular ethnicity only. The model should therefore be trained using a more diverse dataset that adequately captures the wide myriad of human population. In doing so we expect that there will be more challenges such as scalability issue for actual deployment, long training time, handling model bias towards images of particular groups and mode collapse, which is a well known issue for GANs.

An extensive study can be undertaken with regards to the number of layers in the networks, different optimizers and hyperparameter tuning, different network architecture altogether (for eg using InceptionNet [20] for the discriminator) The performance of our model can be improved by training and testing it in cross dataset settings because of which the model is likely to gain even better generalization capability.

When compared to appearance data, motion data can be more useful in separating spoofing faces from real-life faces. Experiments have shown that classification performance can be improved about 10% by using motion information. Hence in the future, considering motion information (using optical flow) along with appearance information to train the model can be explored.

Bibliography

[1] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[2] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[3] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, 2011, pp. 43–58.

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[5] G. Kim, S. Eum, J. K. Suhr, D. I. Kim, K. R. Park, and J. Kim, "Face liveness detection based on texture and frequency analyses," in *2012 5th IAPR international conference on biometrics (ICB)*. IEEE, 2012, pp. 67–72.

[6] J. Li, Y. Wang, T. Tan, and A. K. Jain, "Live face detection based on the analysis of fourier spectra," in *Biometric technology for human identification*, vol. 5404. International Society for Optics and Photonics, 2004, pp. 296–303.

[7] W. Bao, H. Li, N. Li, and W. Jiang, "A liveness detection method for face recognition based on optical flow field," in *2009 International Conference on Image Analysis and Signal Processing*. IEEE, 2009, pp. 233–236.

[8] S. Kim, S. Yu, K. Kim, Y. Ban, and S. Lee, "Face liveness detection using variable focusing," in *2013 International Conference on Biometrics (ICB)*. IEEE, 2013, pp. 1–6.

[9] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face anti-spoofing based on color texture analysis," in *2015 IEEE international conference on image processing (ICIP)*. IEEE, 2015, pp. 2636–2640.

[10] J. Yang, Z. Lei, and S. Z. Li, "Learn convolutional neural network for face anti-spoofing," *arXiv preprint arXiv:1408.5601*, 2014.

[11] H. Li, P. He, S. Wang, A. Rocha, X. Jiang, and A. C. Kot, "Learning generalized deep feature representation for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2639–2652, 2018.

[12] Y. A. U. Rehman, L.-M. Po, M. Liu, Z. Zou, W. Ou, and Y. Zhao, "Face liveness detection using convolutional-features fusion of real and deep network generated face images," *Journal of Visual Communication and Image Representation*, vol. 59, pp. 574–582, 2019.

[13] A. Jaiswal, S. Xia, I. Masi, and W. AbdAlmageed, "Ropad: Robust presentation attack detection through unsupervised adversarial invariance," in *2019 International Conference on Biometrics (ICB)*. IEEE, 2019, pp. 1–8.

[14] L. Li, Z. Xia, X. Jiang, F. Roli, and X. Feng, "Face presentation attack detection in learned color-liked space," *arXiv preprint arXiv:1810.13170*, 2018.

[15] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[16] H. Li, W. Li, H. Cao, S. Wang, F. Huang, and A. C. Kot, "Unsupervised domain adaptation for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1794–1809, 2018.

[17] S. Zhang, X. Wang, A. Liu, C. Zhao, J. Wan, S. Escalera, H. Shi, Z. Wang, and S. Z. Li, "A dataset and benchmark for large-scale multi-modal face anti-spoofing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 919–928.

[18] A. Anjos and S. Marcel, "Counter-measures to photo attacks in face recognition: A public database and a baseline," in *2011 International Joint Conference on Biometrics (IJCB)*, 2011, pp. 1–7.

[19] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *2012 BIOSIG - Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG)*, 2012, pp. 1–7.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.