Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Anti-Spoofing Detection System for Online Examination Real Time Monitoring and Overcoming Fake Bio-Metrics

[1] D.Amritha, [2]S. Haritha,
[1]dhanabalan3amritha@gmail.com[2]harithasatheyamoorthy@gmail.com
Department of Computer Science and Engineering
Prince Dr. K. Vasudevan College of Engineering and Technology, Tamil nadu, India.

ABSTRACT:

In the last years, face recognition systems have gained interest due to face's rich features that offer a strong biometric to recognize individuals for a wide variety of applications. Despite the great deal of progress in facial recognition systems, vulnerabilities to face spoof attacks are mainly overlooked. Though several face anti-spoofing or liveliness detection methods (which determine at the time of capture whether a face is live or spoof) have been proposed, the issue is still unsolved. However, face spoofing attacks remain a problem due to difficulties in finding discriminating and computationally inexpensive features and techniques for spoof recognition. The proposed system a novel method to find the difference between the live face or an spoofing face. We proposed a system were we use the face detection algorithm to find a face and apply face recognition to check for its authenticity and then we check for body temperature normality and abnormality using a thermal camera for distinguishing between genuine and spoof faces like 3-D image, mask , photo or a video of the respective person.

Keywords— Anti-spoofing, Face detection, Face recognition, Face spoofing attacks,Neural networks, Thermal sensor.

I. INTRODUCTION:

Cheating in examinations is something that is inevitable. And this issue has been increasing all over tremendously.Online examination or assessment is one of the key areas of education technology. Online examination usage is likely going to increase for various types of examinations, competitive entrance tests, recruitment tests. Online assessment has proved to be cost effective and better way to filter out and identify suitable candidates.Security is one of the key areas of online assessment which still needs some improvements . Secure online assessment can be useful to prevent all kinds of malpractices tried by the candidates during online exam process.

Over the past few years, biometric face spoofing attack have increased significantly all across the globe.Nowadays, hackers are trying everything possible (from 3D mask to printed photos) to bypass biometric face authentication systems.The study of the vulnerabilities of biometric systems against spoofing has been a very active field of research in recent years. Biometric spoofing, also referred to as biometric direct attacks, is widely understood in the specialised literature as the ability to fool a biometric system into recognizing an illegitimate user as the genuine one, by means of presenting to the sensor a synthetic forged version of the original biometric trait.Facial anti-spoofing is the task of preventing false facial verification by using a photo, video,

mask or a different substitute for an authorized person's face. In 2D-face recognition spoofing attacks are generally carried out in one of three ways-

- Photo Attacks: These fraudulent access attempts are performed presenting to the recognition system a photograph of the genuine user. This image may be printed on a paper or displayed on the screen of a digital device such as a mobile phone or a tablet.

- Video Attacks: In this case, the attacker does not use a still image, but replays a video of the genuine client using a digital device say for e.g., mobile phone, tablet or laptop. A more sophisticated way to trick the system, which usually requires a looped video of a victim's face. This approach ensures behaviour and facial movements to look more 'natural' compared to holding someone's photo.

- Mask Attacks: In these cases, the spoofing artefact is a 3D mask of the genuine client's face, increasing the difficulty to find accurate countermeasures. During this type of attack, a mask is used as the tool of choice for spoofing.It's an even more sophisticated attack than playing a face video.

In addition, all the previous types of attacks have a number of variants depending on the resolution of the attack device, the type of support used to present the fake copy, or the type of external variability allowed. All these attacks have been shown in different works to pose a real security threat to 2-D face recognition systems.The previous efforts, and other similar works, have led to big advances in the field of security-enhancing technologies for face-based applications. However, in spite of this noticeable improvement, the development of efficient protection methods against known direct attacks has proven to be a challenging task that still requires novel algorithms.

DEEP LEARNING:

Deep learning is a computer software that imitates the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural networks.Deep learning algorithms are constructed with connected layers.

- The first layer is called the Input Layer.
- The last layer is called the Output Layer.
- All layers in between are called Hidden Layers.

Each Hidden layer is composed of neurons. The neurons are connected to each other. The neuron will process and then propagate the input signal it receives the layer above it. The strength of the signal given the neuron in the next layer depends on the weight, bias and activation function.The network consumes large amounts of input data and operates them through multiple layers; the network can learn increasingly complex features of the data at each layer.

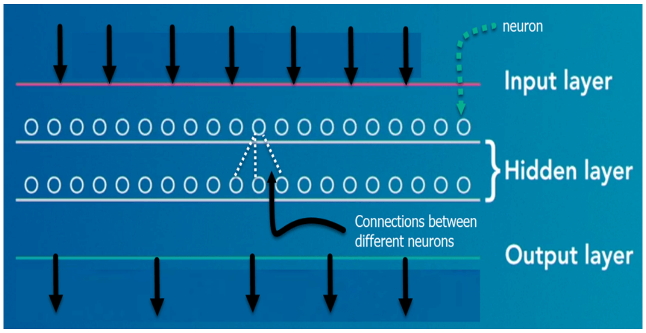


Fig. 1 Layers in deep learning

Each layer represents a deeper level of knowledge, i.e., the hierarchy of knowledge. A neural network with four layers will learn more complex feature than with that with two layers.The learning occurs in two phases.

- The first phase consists of applying a nonlinear transformation of the input and create a statistical model as output.
- The second phase aims at improving the model with a mathematical method known as derivative.

The neural network repeats these two phases hundreds to thousands of time until it has reached a tolerable level of accuracy. The repeat of this two-phase is called an iteration.

CONVOLUTIONAL NEURAL NETWORK(CNN):

Convolutional neural networks are made of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts, are mathematical functions that calculate the weighted sum of multiple inputs and outputs an activationvalue.

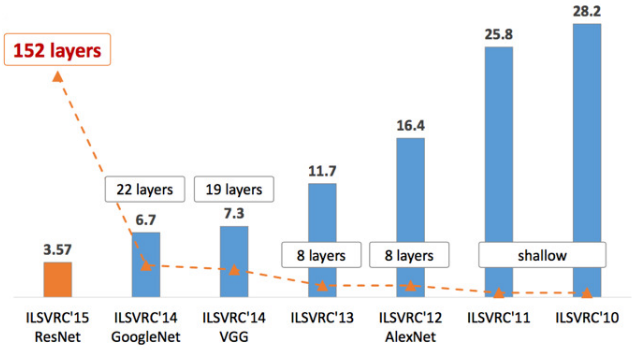


Fig. 2 Types of CNN

The behavior of each neuron is defined by its weights. When fed with the pixel values, the artificial neurons of a CNN pick out various visual features.When you input an image into a ConvNet, each of its layers generates several activation maps. Activation maps highlight the relevant features of the image. Each of the neurons takes a patch of pixels as input, multiplies their color values by its weights, sums them up, and runs them through the activation function.The first (bottom) layer of the CNN basically detects basic features such as horizontal, vertical, and diagonal edges. The output of the first layer is fed as input of the next layer, which extracts more complex features, such as corners and combinations of edges. As you move deeper into the convolutional neural network, the layers start detecting higher-level features such as objects, faces, and more.

II. RELATED WORK:

[1] This articleuseda labelled eyeblink in the wild dataset (i.e., HUST-LEBW) of 673 eyeblink video samples (i.e., 381 positives, and 292 negatives).Then, we formulate eyeblink detection task as a binary spatial-temporal pattern recognition problem. After locating and tracking human eyes using Seeta Face engine and KCF (Kernelized Correlation Filters) tracker respectively, a modified LSTM model able to capture the multi-scale temporal information is proposed to verify eyeblink. A feature extraction approach that reveals the appearance and motion characteristics simultaneously is also proposed. [2] Here they consider the task of face anti-spoofing as the detection of SMCs from the image. We propose and train a Contour Enhanced Mask R-CNN (CEM-RCNN) model for the detection. This model detects the existence of the SMCs by incorporating the contour objectiveness which measures how likely an object contains the SMCs. [3] This paper present a systematic study of the vulnerability of automatic speaker verification to a diverse range of spoofing attacks. It starts with a thorough analysis of the spoofing effects of five speech synthesis and eight voice conversion systems, and the vulnerability of three speaker verification systems under those attacks. Then introduced a number of countermeasures to prevent spoofing attacks from both known and unknown attackers. Finally, they benchmark

automatic systems against human performance on both speaker verification and spoofing detection tasks.[4] This article proposed State-of-the-art spoof detection methods tend to overfit to the spoof types seen during training and fail to generalize to unknown spoof types. They propose a face anti-spoofing framework, namely Self-Supervised Regional Fully Convolutional Network (SSR-FCN), that is trained to learn local discriminative cues from a face image in a self-supervised manner. The proposed framework improves generalizability while maintaining the computational efficiency of holistic face anti-spoofing approaches. The proposed method is interpretable since it localizes which parts of the face are labeled as spoofs. [5]The paper introduce an unsupervised domain adaptation face anti-spoofing scheme to address the real world scenario that learns the classifier for the target domain based on training samples in a different source domain. In particular, an embedding function is first imposed based on source and target domain data, which maps the data to a new space where the distribution similarity can be measured. Subsequently, the Maximum Mean Discrepancy between the latent features in source and target domains is minimized such that a more generalized classifier can be learned. State-of-the-art representations including both hand-crafted and deep neural network learned features are further adopted into the framework to quest the capability of them in domain adaptation. Moreover, they introduce a new database for face spoofing detection, which contains more than 3000 face samples with a large variety of spoofing types, capture devices, illuminations, etc.

III. METHODS AND MATERIALS:

In the proposed system, we will be monitoring the face of a person and will distinguish it between a real face and spoof face by using a thermal camera.We will be checking the temperature of the human face, which will differ from all otheridentities like masks, 3-d images, photos etc., and will produceresults based on the temperature recorded in the thermal camera.Experimental analysis has been done in real time and it shows a promising result. The real time execution of this project reduces spoofing attacks of face in minimal amount of time and can be implemented at low cost and provides great results. It can automatically detect the human face and distinguish it between real and spoof and alerts if spoofing occurs.

METHODOLOGY:

In this project we will be collecting the images of the authorized people for training. The collected data will be labelled according to the names of the required person. Then images are augmented by using image generator, augmentation method means images are rotated in all angles and they are loaded for training. In training we will be extracting features of the faces and we will save them in a pickle file. We will now implementing this into real time camera. Then face detection is applied to the camera frame. The “Ultra-Light-Fast-Generic-Face-Detector” is designed for general-purpose face detection applications in low-power computing devices and is applicable to both Android and iOS phones as well as PCs (CPU and GPU).When a face appears in front of the camera, An ROI of the face is extracted and then we will be comparing the features with the trained pickle file and the registered face will be labelled as per the face recognized. For Face Recognition module here we using Mobile face net algorithm. Now a thermal camera detects the middle point of the face and then records the nearest thermal temperature and stores it in an array.The AMG8833 thermal imaging camera sensor is an 8x8 infrared thermal sensor array. It has a temperature measurement range of 0°C to 80°C (32°F to 176°F).When connected to the laptop processor, it will return a set of 64 separate infrared temperature readings via I2C. If the temperature values are in the range of 37 Celsius to 47 Celsius, then the system continues confirms it as a genuine face. If the temperature values are out of range then, the system confirms it as a spoofed face. A genuine face is now marked present in the particular period. If a face is detected as a spoofed face it will not bemarked present even if the face is registered.

Ultra-Light-Fast-Generic-Face-Detector:

Ultra-Light-Fast-Generic-Face-Detectormodel is a lightweight facedetection model designed for edge computing devices.In terms of model size, the default FP32 precision (.pth) file size is 1.04~1.1MB, and the inference framework int8 quantization size is about 300KB.In terms of the calculation amount of the model, the input resolution of 320x240 is about 90~109 MFlops.

Mobile FaceNet Algorithm:

FaceNet is a face recognition system developed in 2015 by researchers at Google that achieved then state-of-the-art results on a range of face recognition benchmark datasets. The FaceNet system can be used broadly thanks to multiple third-party open source implementations of the model and the availability of pre-trained models.The FaceNet system can be used to extract high-quality features from faces, called face embeddings, that can then be used to train a face identification system.

BATCH COLLECTION:

A data set is a collection of data. In other words, a data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable, and each row corresponds to a given member of the data set in question. In Deep Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions.From training, tuning, model selection to testing, we use different data sets: the training set, and the testing set. The training data set is the one used to train an algorithm to understand how to apply concepts such as neural networks, to learn and produce results. It includes both input data and the expected output. The test data set is used to evaluate how well your algorithm was trained with the training data set. Ten images of each students are captured and uploaded in real time as dataset for training purposes.

ANTI-SPOOFING:

The AMG8833 thermal imaging camera sensor is an 8x8 infrared thermal sensor array.It has a temperature measurement range of 0°C to 80°C (32°F to 176°F).When connected to the laptop processor ,it will return a set of 64 separate infrared temperature readings via I2C. It is it's compact and simple, and easy to integrate. The sensor only supports I2C and has a configurable interrupt pin that can be triggered when any single pixel is above or below the threshold being set.

FACE DETECTION:

The “Ultra-Light-Fast-Generic-Face-Detector” is designed for general-purpose face detection applications in low-power computing devices and is applicable to both Android and iOS phones as well as PCs (CPU and GPU). It is 22-layers deep neural network that directly trains its output to be a 128-dimensional embedding. The loss function used at the last layer is called triplet loss. The model is a real-time ultra-lightweight universal face detection model designed for edge computing devices or low-power devices. It can be used in low-power computing devices such as ARM/intel processor for real-time common scene faces.

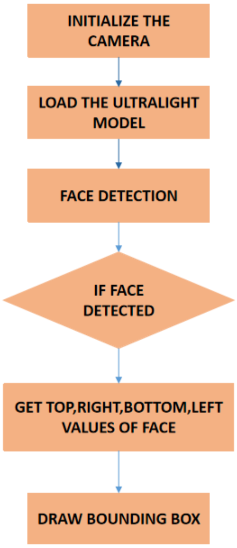


Fig. 3 face detection

MOBILE FACENET FACE RECOGNITION:

For Face Recognition module here we using Mobile facenet algorithm. Mobile Face Nets, which use less than 1 million parameters and are specifically tailored for high-accuracy real-time face verification on mobile and embedded devices. Mobile Face Nets achieve significantly superior accuracy as well as more than 2 times actual speedup over MobileNetV2. The fastest one of MobileFaceNets has an actual inference time of 18 milliseconds on a mobile phone. For face verification, MobileFaceNets achieve significantly improved efficiency over previous state-of-the-art mobile CNNs.

FAKE FACE DETECTION:

The first step in the system is to trigger the thermal sensor and then initiate the thermal sensor. The next step is to get the thermal values of the face in an array. Detect the middle point from the face. Find the nearest visible thermal value and store it in an array. Output the temperature. If the recorded thermal values are not in the range 34 C- 42 C, possible spoofing is detected. If the temperature values are normal to that of a human body then the user has been authenticated.

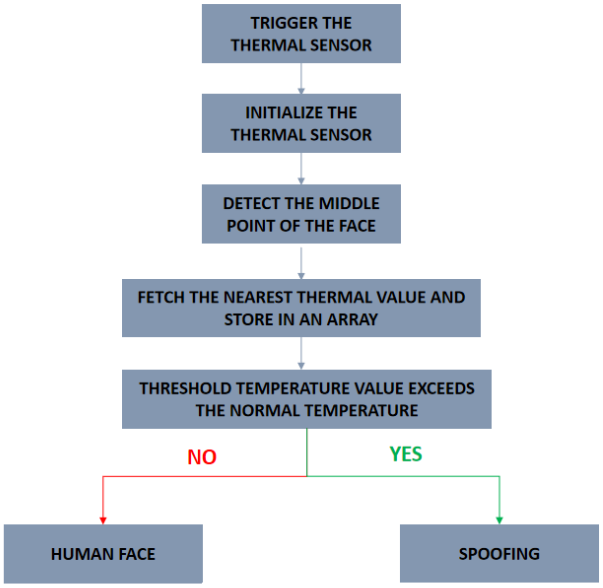

TRIGGER THE THERMAL SENSOR
INITIALIZE THE THERMAL SENSOR
DETECT THE MIDDLE POINT OF THE FACE
FETCH THE NEAREST THERMAL VALUE AND STORE IN AN ARRAY
THRESHOLD TEMPERATURE VALUE EXCEEDS THE NORMAL TEMPERATURE
NO
YES
HUMAN FACE
SPOOFING

Fig. 4 Fake face detetction

TECHNOLOGY USED:

EMBEDDED SYSTEM:

An embedded system is a microprocessor-based computer hardware system with the software that is designed to perform a dedicated function, either as an independent system or as a part of a large system. At the core is integrated circuit designed to carry out computation for real-time operations.Embedded systems are managed by microcontrollers or digital signal processors (DSP), application-specific integrated circuits (ASIC), field-programmable gate arrays (FPGA), GPU technology and gate arrays. These processing systems are integrated with components dedicated to handling electric and/or mechanical interfacing.

OpenCV:

OpenCV is the trending open source library for computer vision, image processing and machine learning, and now features GPU acceleration for real-time operation.Currently OpenCV supports a huge variety of programming languages like C++, Python, Java, etc., and is available on different platforms including Windows, Linux, OS X, Android, iOS etc. Also, interfaces based on CUDA and OpenCL are also under active development for high-speed GPU operations.OpenCV-Python is the Python API of OpenCV. It combines the best qualities of OpenCV C++ API and Python language.

SYSTEM ARCHITECTURE:

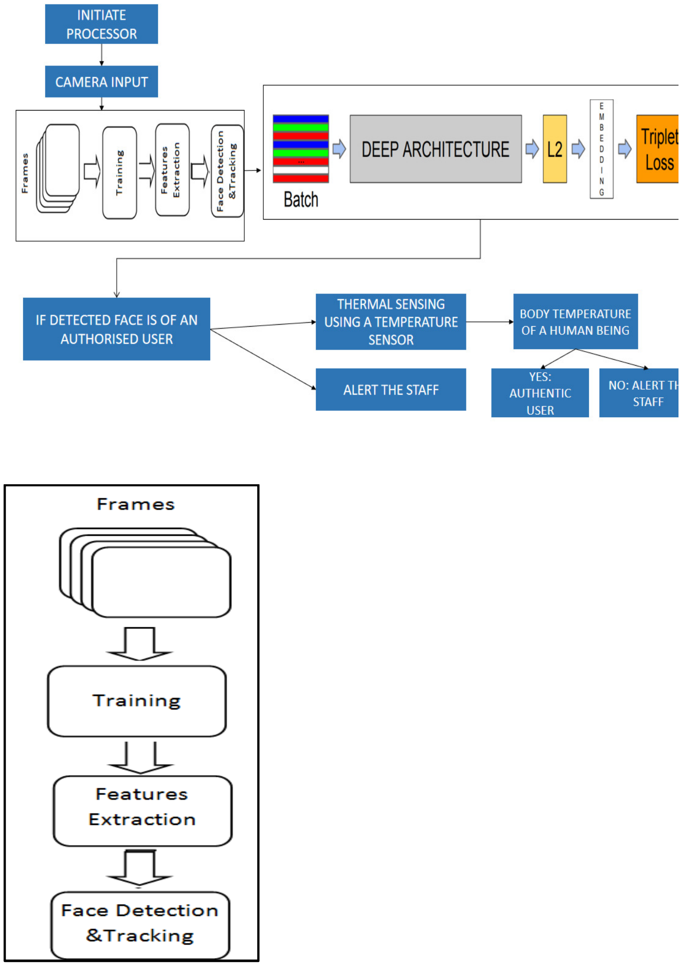


Fig. 5 System Architecture.

IV. RESULTS AND DISCUSSION:

PERFORMANCE MEASURE:

FEATURES	EXISTING SYSTEM	PROPOSED SYSTEM
Number of Faces detection (Given 150 faces)	70	140
Time taken for detection	3 seconds	200 milli seconds
Model Size	20MB	1.04~1.1MB
Efficiency	Slow and inefficient	Fast and efficient
Spoofing	Spoofing of face is possible	Spoofing is not at all possible
Module used to identify spoofing	Eye blink is detected	Thermal sensors

OUTPUT SCREENSHOTS:

DATASET COLLECTION

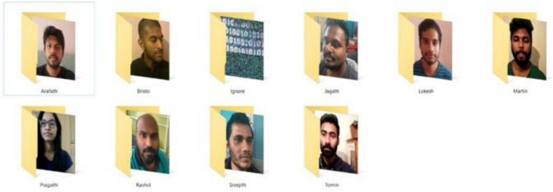
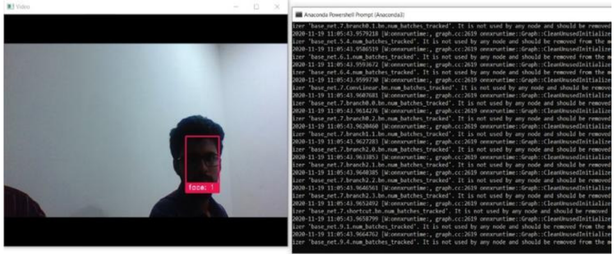


Fig. 6 Dataset collection

FACE DETECTION



```
izer 'base_net.7.branch0.1.bn.num_batches_tracked'. It is not used by any node and should be removed
2020-11-19 11:05:43.9579218 [W:onnxruntime:, graph.cc:2619 onnxruntime::Graph::CleanUnusedInitialize
izer 'base_net.5.4.num_batches_tracked'. It is not used by any node and should be removed from the m
2020-11-19 11:05:43.9586519 [W:onnxruntime:, graph.cc:2619 onnxruntime::Graph::CleanUnusedInitialize
izer 'base_net.6.1.num_batches_tracked'. It is not used by any node and should be removed from the m
2020-11-19 11:05:43.9593672 [W:onnxruntime:, graph.cc:2619 onnxruntime::Graph::CleanUnusedInitialize
izer 'base_net.6.4.num_batches_tracked'. It is not used by any node and should be removed from the m
2020-11-19 11:05:43.9599730 [W:onnxruntime:, graph.cc:2619 onnxruntime::Graph::CleanUnusedInitialize
izer 'base_net.7.ConvLinear.bn.num_batches_tracked'. It is not used by any node and should be remove
2020-11-19 11:05:43.9607681 [W:onnxruntime:, graph.cc:2619 onnxruntime::Graph::CleanUnusedInitialize
izer 'base_net.7.branch0.0.bn.num_batches_tracked'. It is not used by any node and should be removed
2020-11-19 11:05:43.9614276 [W:onnxruntime:, graph.cc:2619 onnxruntime::Graph::CleanUnusedInitialize
izer 'base_net.7.branch0.2.bn.num_batches_tracked'. It is not used by any node and should be removed
2020-11-19 11:05:43.9620460 [W:onnxruntime:, graph.cc:2619 onnxruntime::Graph::CleanUnusedInitialize
izer 'base_net.7.branch1.1.bn.num_batches_tracked'. It is not used by any node and should be removed
2020-11-19 11:05:43.9627283 [W:onnxruntime:, graph.cc:2619 onnxruntime::Graph::CleanUnusedInitialize
izer 'base_net.7.branch2.0.bn.num_batches_tracked'. It is not used by any node and should be removed
2020-11-19 11:05:43.9633853 [W:onnxruntime:, graph.cc:2619 onnxruntime::Graph::CleanUnusedInitialize
izer 'base_net.7.branch2.1.bn.num_batches_tracked'. It is not used by any node and should be removed
2020-11-19 11:05:43.9640385 [W:onnxruntime:, graph.cc:2619 onnxruntime::Graph::CleanUnusedInitialize
izer 'base_net.7.branch2.2.bn.num_batches_tracked'. It is not used by any node and should be removed
2020-11-19 11:05:43.9646561 [W:onnxruntime:, graph.cc:2619 onnxruntime::Graph::CleanUnusedInitialize
izer 'base_net.7.branch2.3.bn.num_batches_tracked'. It is not used by any node and should be removed
2020-11-19 11:05:43.9652492 [W:onnxruntime:, graph.cc:2619 onnxruntime::Graph::CleanUnusedInitialize
izer 'base_net.7.shortcut.bn.num_batches_tracked'. It is not used by any node and should be removed
2020-11-19 11:05:43.9658799 [W:onnxruntime:, graph.cc:2619 onnxruntime::Graph::CleanUnusedInitialize
izer 'base_net.9.1.num_batches_tracked'. It is not used by any node and should be removed from the m
2020-11-19 11:05:43.9664762 [W:onnxruntime:, graph.cc:2619 onnxruntime::Graph::CleanUnusedInitialize
izer 'base_net.9.4.num_batches_tracked'. It is not used by any node and should be removed from the m
```

Fig. 7 Face Detection

SAVED FACE:

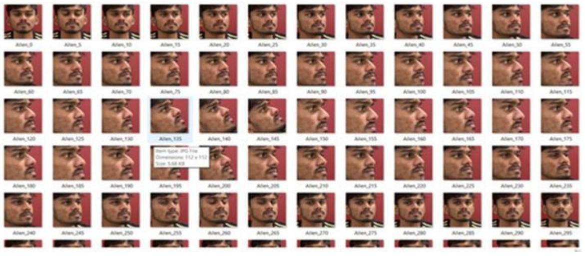
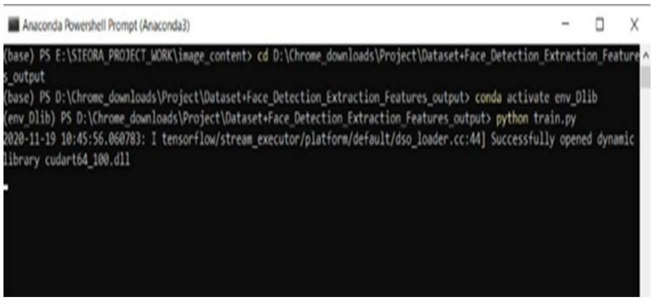


Fig. 8 Saved face

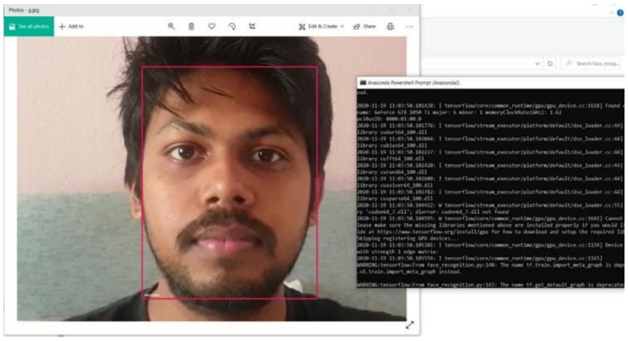
FACE FEATURE EXTRACTION:



```
(base) PS E:\SIEORA_PROJECT_WORK\image_content> cd D:\Chrome_downloads\Project\Dataset+Face_Detection_Extraction_Feature
s_output
(base) PS D:\Chrome_downloads\Project\Dataset+Face_Detection_Extraction_Features_output> conda activate env_Dlib
(env_Dlib) PS D:\Chrome_downloads\Project\Dataset+Face_Detection_Extraction_Features_output> python train.py
2020-11-19 10:45:56.060783: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic
library cudart64_100.dll
```

Fig. 9 Feature extraction

FACE RECOGNITION:



```
ead.

2020-11-19 11:03:50.101438: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1618] Found
name: GeForce GTX 1050 Ti major: 6 minor: 1 memoryClockRate(GHz): 1.62
pciBusID: 0000:01:00.0
2020-11-19 11:03:50.101776: I tensorflow/stream_executor/platform/default/dso_loader.cc:44]
library cudart64_100.dll
2020-11-19 11:03:50.102044: I tensorflow/stream_executor/platform/default/dso_loader.cc:44]
library cublas64_100.dll
2020-11-19 11:03:50.102237: I tensorflow/stream_executor/platform/default/dso_loader.cc:44]
library cufft64_100.dll
2020-11-19 11:03:50.102420: I tensorflow/stream_executor/platform/default/dso_loader.cc:44]
library curand64_100.dll
2020-11-19 11:03:50.102600: I tensorflow/stream_executor/platform/default/dso_loader.cc:44]
library cusolver64_100.dll
2020-11-19 11:03:50.102782: I tensorflow/stream_executor/platform/default/dso_loader.cc:44]
library cusparse64_100.dll
2020-11-19 11:03:50.104412: W tensorflow/stream_executor/platform/default/dso_loader.cc:55]
ry 'cudnn64_7.dll'; dlerror: cudnn64_7.dll not found
2020-11-19 11:03:50.104595: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1641] Cannot
lease make sure the missing libraries mentioned above are installed properly if you would li
ide at https://www.tensorflow.org/install/gpu for how to download and setup the required lib
Skipping registering GPU devices...
2020-11-19 11:03:50.105381: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1159] Device
with strength 1 edge matrix:
2020-11-19 11:03:50.105559: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1165]
WARNING:tensorflow:From face_recognition.py:140: The name tf.train.import_meta_graph is depr
.v1.train.import_meta_graph instead.

WARNING:tensorflow:From face_recognition.py:143: The name tf.get_default_graph is deprecated
```

Fig. 10 Face recognition

V. CONCLUSION AND FUTURE ENHANCEMENT:

The project is successfully implemented to prevent spoofed faces from being detected by face detection systems for genuine faces. The project can save properties and private data from being compromised due to unauthorized access or members. The project eliminates the chances of an error in face detection and recognition systems. On successful completion of project we can implement this as a real time product. Further we can implement this for access control systems where spoofing has become common.

VI. REFERENCES:

[1] Guilei Hu, Yang Xiao , Zhiguo Cao , Lubin Meng, Zhiwen Fang , Joey Tianyi Zhou , and Junsong Yuan , Senior Member, IEEE,"Towards Real-Time Eyeblink Detection in the Wild: Dataset, Theory and Practices",2020.

[2] Xun Zhu, Sheng Li, Member, IEEE, Xinpeng Zhang, Haoliang Li and Alex C. Kot, Fellow, IEEE,"Detection of Spoofing Medium Contours for Face Anti-spoofing",2019.

[3] Zhizheng Wu□ , Phillip L. De Leon, Senior Member, IEEE, Cenk Demiroglu, Ali Khodabakhsh, Simon King, Fellow IEEE, Zhen-Hua Ling, Daisuke Saito, Bryan Stewart, Tomoki Toda, Mirjam Wester, and Junichi Yamagishi, Senior Member, IEEE,"Anti-Spoofing for Text-Independent Speaker Verification: An Initial Database, Comparison of Countermeasures, and Human Performance",2016.

[4] Debian deb,Anil K Jain,"Look Locally Infer Globally: A Generalizable Face Anti-Spoofing Approach",2017.

[5] Haoliang Li, Wen Li, Hong Cao, Senior Member, IEEE, Shiqi Wang, Member, IEEE, Feiyue Huang, and Alex C. Kot, Fellow, IEEE,"Unsupervised Domain Adaptation for Face Anti-Spoofing",2018.

[6] Wenyun Sun , Member, IEEE, Yu Song , Member, IEEE, Changsheng Chen , Member, IEEE, Jiwu Huang , Fellow, IEEE, and Alex C. Kot , Fellow, IEEE,"Face Spoofing Detection Based on Local Ternary Label Supervision in Fully Convolutional Networks",Vol 15 2020.

[7] Jon Sanchez, Ibon Saratxaga, Inma Hernáez, Eva Navas, Daniel Erro, and Tuomo Raitio,"Toward a Universal Synthetic Speech Spoofing Detection Using Phase Information",VOL. 10, NO. 4, APRIL 2015.

[8] Alejandro Gomez-Alanis , Antonio M. Peinado , Senior Member, IEEE, Jose A. Gonzalez , and Angel M. Gomez,"A Gated Recurrent Convolutional Neural Network for Robust Spoofing Detection",2017.

[9] Tomi Kinnunen , Member, IEEE, Héctor Delgado , Member, IEEE, Nicholas Evans, Member, IEEE, Kong Aik Lee , Senior Member, IEEE, Ville Vestman , Andreas Nautsch , IEEE, Massimiliano Todisco, IEEE, Xin Wang , Member, IEEE, Md Sahidullah , IEEE, Junichi Yamagishi , Senior Member, IEEE, and Douglas A. Reynolds , Fellow, IEEE,"Tandem Assessment of Spoofing Countermeasures and Automatic Speaker Verification: Fundamentals",2020.

[10] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context- aware attention LSTM networks for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 7, Jul. 2017, p.43.

[11] Peng Zhang2,3 , Fuhao Zou1□ , Zhiwen Wu2 , Nengli Dai3 Skarpness Mark2 , Michael Fu2 , Juan Zhao2 , Kai Li1, "FeatherNets: Convolutional Neural Networks as Light as Feather for Face Anti-spoofing".

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In ´ Proceedings of the IEEE international conference on computer vision, pages 2980–2988, 2017

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision, pages 1026–1034, 2015.

[14] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[15] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In Proceedings of the European Conference on Computer

Vision (ECCV), pages 534–551, 2018.

[16] Chi Nhan Duong, Kha Gia Quach, Ngan Le, Nghia Nguyen, and Khoa Luu. Mobiface: A lightweight deep learning face recognition on mobile devices. arXiv preprint arXiv:1811.11080, 2018.

[17] Zheng Qin, Zhaoning Zhang, Xiaotao Chen, Changjian Wang, and Yuxing Peng. Fd-mobilenet: Improved mobilenet with a fast downsampling strategy. In 2018 25th IEEE International Conference on Image Processing (ICIP), pages 1363–1367. IEEE, 2018.

[18] ie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7132–7141, 2018.