IJNRD.ORG ISSN : 2456-4184

INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG

An International Open Access, Peer-reviewed, Refereed Journal

USES OF DEEP LEARNING FOR FAKE IMAGE DETECTION

B.Venkata Vamsi Krishna[#1], Kammari Ryagnirvesh[#2], Potharaju Mahesh [#3], K.Shilpa(Assistant Professor) [#4]

[#] *Dept. Of CSE, CMR Technical Campus*

***Abstract*— Biometric technologies are useful now for identifying people, but criminals alter their look, behaviour, and psychological makeup to trick identification systems. We are employing a novel method called Deep Texture Features extraction from photos to solve this issue, followed by the construction of a machine learning model using the CNN (Convolution Neural Networks) algorithm. This method is also known as LBPNet or NLBPNet since it relies so heavily on the LBP (Local Binary Pattern) algorithm for features extraction.**

In order to identify false face photos, we are proposing an LBP-based machine learning convolution neural network dubbed LBPNET. Here, we will first extract LBP from the photos, and then we will train the convolution neural network on the LBP descriptor images to produce the training model. Every time a new test picture is uploaded, the training model will use that image to determine if the test image contains fraudulent images or not. Listed below are some LBP details.

***Keywords*—**Convolutional Neural Network (CNN), Local Binary Patterns (LBP), Deep Learning, detection of fake images

1. INTRODUCTION

Both active schemes and passive schemes are often utilised in the conventional picture forgery detection technique, however they both fall short when applied to GANs. The externally inserted signal (i.e., watermark) is integrated in the source picture without visible artefacts in the active schemes. A watermark extraction procedure is carried out on the target picture to recover the watermark in order to determine if the image has been altered or not. The target picture's tampered areas can be located or found using the extracted watermark image.

The active image forgery detector cannot be used to extract the watermark picture since there is no "source image" for the images created by GANs. The original image's statistical data is used by the passive image forgery detector to distinguish it from counterfeit pictures. This quality makes it possible to identify the false section in the image using the intrinsic statistical data. Unfortunately, because GANs are created from a low-dimensional random vector, the passive image forgery detector cannot be utilised to detect the fraudulent picture produced by GANs. The false picture is not altered from its original image, hence nothing changes in the image that GANs create.

Our LBP-based Deep Learning Convolutional Neural Network, dubbed LBPNET in this research, was created to learn and execute bitwise operations end-to-end. In order to identify false face pictures, LBPNet employs local binary comparisons and random projections, which is a significant way to increase memory efficiency and inference speed. To create a training model, we first extract LBP from photos and then train LBP descriptor images with convolution neural networks. Test pictures are then employed to evaluate the training's correctness and establish whether or not they are fraudulent.

2.1 OBJECTIVE

In order to develop a convolutional neural network that can discriminate between false and real photos, this project makes use of a variety of Deep Learning methods and methodologies. Our ultimate objective with this research is to create a robust system that can quickly identify any type of digital forgeries, hence minimising any potentially detrimental outcomes that may otherwise result from the synthesis of false digital material. At the same time, reducing the aforementioned artefacts and extracting characteristics that are unaffected by factors like lighting, posture, emotions, overlapping objects, etc. are necessary for facial recognition. Our research's objective is to create a technique for identifying phoney faces from a single image.

One of the most well-known biometric identity authentication techniques is face recognition, which is frequently used in the security of businesses and organisations as well as public safety in areas like stadiums, train stations, and airport terminals. It is also frequently used for outdoor surveillance. Gabor filters and Local Binary Patterns (LBPs) are two methods for detecting local features, while methods for detecting generalised features (principal component analysis, Bayesian classification, and metric models) and deep learning techniques are methods for detecting generalised features. Research in this area started in the 1990s. Deep learning-based facial recognition now has a 99.80% accuracy rate. At the same time, 97.53% precision is said to be displayed by human eyesight.

Face recognition systems must have a false face detection module since it is quite simple to alter a face image or provide a brief video imitating another person. Typically, the visual processing module and the recognition module are introduced before the false face detection and alignment module. It's important to remember that the target functions of fake face detection and face recognition are distinct. Finding evidence of the "liveliness" of the face is linked to the detection of forgery. Lighting, shadows, glare, scene depth, etc., are therefore crucial.

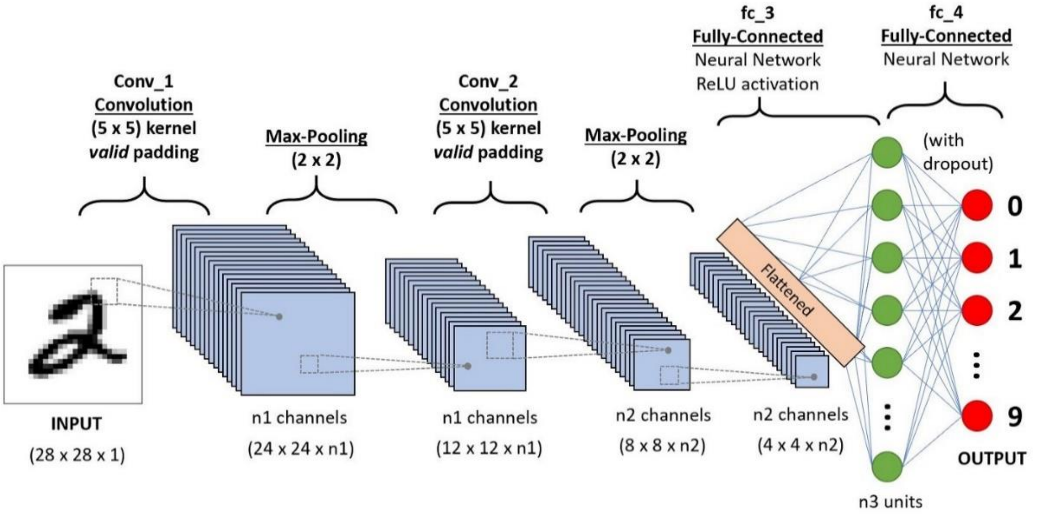


Figure 1: Operation of CNN

2.2 FEATURES

Recurrent neural networks (RNN), convolutional neural networks (CNN), and long short-term memory are used in this work to provide a thorough analysis for deepfake identification (LSTM). Researchers in this field will find this survey informative and advantageous as it provides: 1) a detailed description of the existing research papers; 2) datasets utilised in this field; 3) the shortcomings of the present methodologies and suggestions for future study. The following is a summary of the results of our survey.

Artificial neural networks (ANNs) take some of their fundamental ideas from how the human brain operates. A single input layer, one or more hidden layers, and one output layer make up the several layers of neural networks. A group of input values are used as the input to neural networks. Neural networks are designed to anticipate and categorise such values into predetermined groups.

The input layer is the initial layer of a neural network, and it receives input data and passes them on to the next layer. The input values in our case are x1, x2, x3, and x4. The second layer consists of hidden layers, which are made up of artificial neurons, a group of linked units (nodes).

The edges that connect the neurons represents how all the neurons are interconnected and how can receive and send signals through multi layers. Each connection has a weight associated with it which represents the connections between two units. In our network, the 1st hidden layer consists of 3 neurons and the 2nd layer contains 4 neurons.

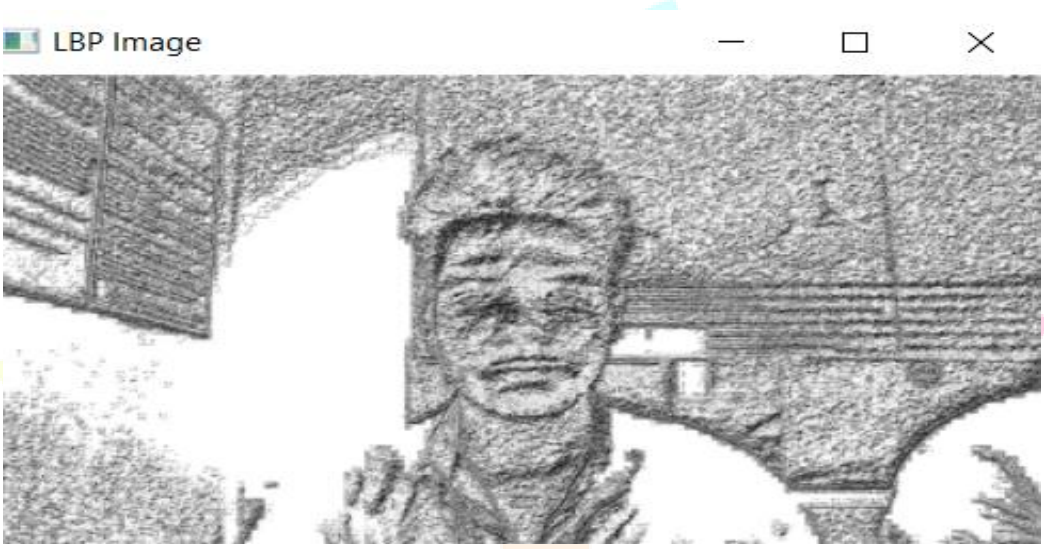


Figure 2: LBP Generated Image

3. LITERATURE SURVEY

Several research teams have run numerous tests throughout the years. Many of the groupings include the following:

▶ [1] Progressive growth of GANs enhanced quality, stability, and variety, according to T. Karras, T. Aila, S. Laine, and J. Lehtinen. a preprint published as arXiv:1710.10196 in 2017256.

Several applications, such as voice synthesis (van den Oord et al., 2016a), picture-to-image translation, and image inpainting, emloy generative techniques that create fresh samples from high-dimensional data distributions, such as images (Iizuka et al., 2017).

At the moment, variational autoencoders (VAE) (Kingma & Welling, 2014), generative adversarial networks (GAN), and autoregressive models are the most popular methods (Goodfellow et al., 2014).

[2] Unpaired image-to-image translation using cycle consistent 259 adversarial networks was developed by J.Y. Zhu, T. Park, P. Isola, and A.A. Efros. 2017 arXiv
Preprint.

On a beautiful spring day in 1873, when Claude Monet set up his easel by the River in Argenteuil (Screenshot 1, top-left), what might he see? If colour photography had been available, it may have captured the clear blue sky and the sparkling river that reflected it. Monet used a vibrant colour scheme and airy brushstrokes to depict the same scenario. Despite never having seen a side-by-side comparison of a Monet painting and a photograph of the scene he created, we can still see all of this.

[3] Learning to recognise false face photos in the wild. C.Hsu, C.Lee, and Y.Zhuang. 2018.

The security concerns resulting from malicious applications of such image production/synthesis have grown in importance with the quick development of deep learning algorithms for picture formation. For instance, the progressive growth of GANs (PGGAN) method put out by nVidia has shown that it is simple to synthesis realistic and high-resolution facial pictures. It can be utilised to fabricate a personal Facebook account in order to defraud anything or someone.

[4]HT Chang and CC Hsu, Image authentication with tampering localisation based on watermark 266 embedding in wavel et domain. 48, 057002, Optical Engineering, 2009.

This project's goal is to recognise false photos. The issue with current false picture detection systems is that they can only b e used to identify particular types of manipulation, such as cutting, colouring, and so on. To solve the issue, we used machi ne learning and neural networks to identify practically all types of picture manipulation.

4. PROPOSED SYSTEM

It is possible to think about fake image detection as a Binary Classification issue that can be solved with supervised learning.The suggested approach overcomes the shortcomings of conventional forgery detection methods via GANs by utilising Deep Learning principles like CNN and LBP Algorithm.A series of training photos is used to train the model.

The test image is provided as input once the CNN model has been created.In addition to classifying the image as authentic or false, the algorithm also produces an LBP reference

image.Both we and the user will incur very little expense in developing and utilising this

System respectively.

A PC or desktop with the necessary software characteristics is the minimum need.

We may classify our solution as a web application.

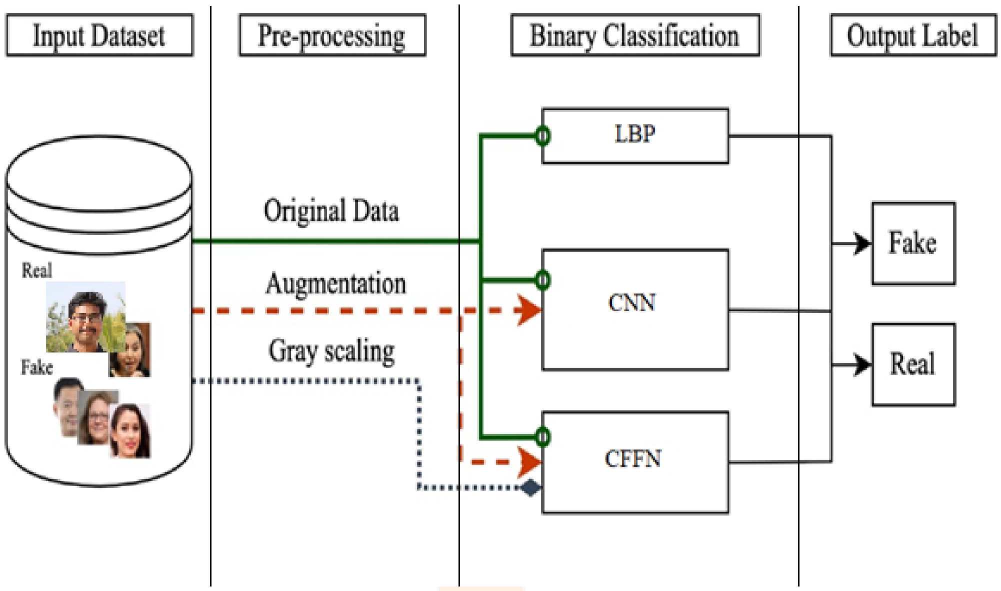
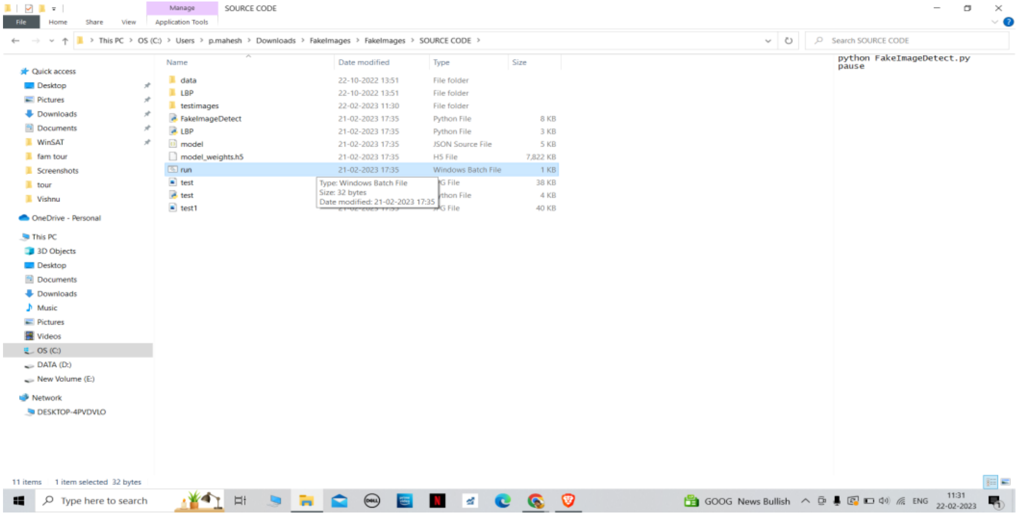


Figure 3: Architecture for Detecting Fake Images

RESULT

Every time we submit a new test picture, the training model will use that image to determine if the test image contains fraudulent images or not.



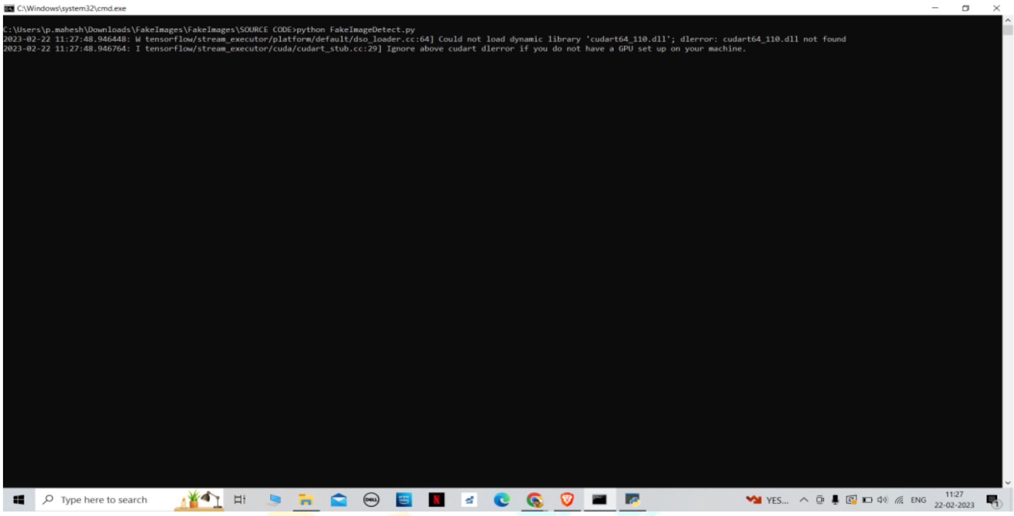
Name	Date modified	Type	Size
data	22-10-2022 13:51	File folder	
LBP	22-10-2022 13:51	File folder	
testimages	22-02-2023 11:30	File folder	
FakeImageDetect	21-02-2023 17:35	Python File	8 KB
LBP	21-02-2023 17:35	Python File	3 KB
model	21-02-2023 17:35	JSON Source File	5 KB
model_weights.h5	21-02-2023 17:35	H5 File	7,822 KB
run	21-02-2023 17:35	Windows Batch File	1 KB
test		G File	38 KB
test		thon File	4 KB
test1		G File	40 KB

Type: Windows Batch File
Size: 32 bytes
Date modified: 21-02-2023 17:35

```
python FakeImageDetect.py
pause
```

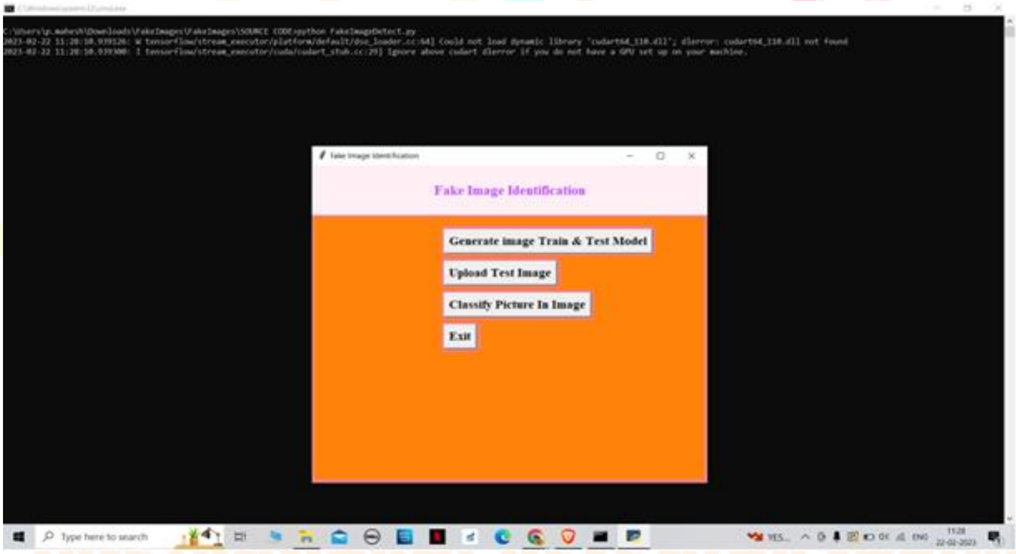
11 items 1 item selected 32 bytes

Screenshot 1: Running the Program Of Fake Image Detection



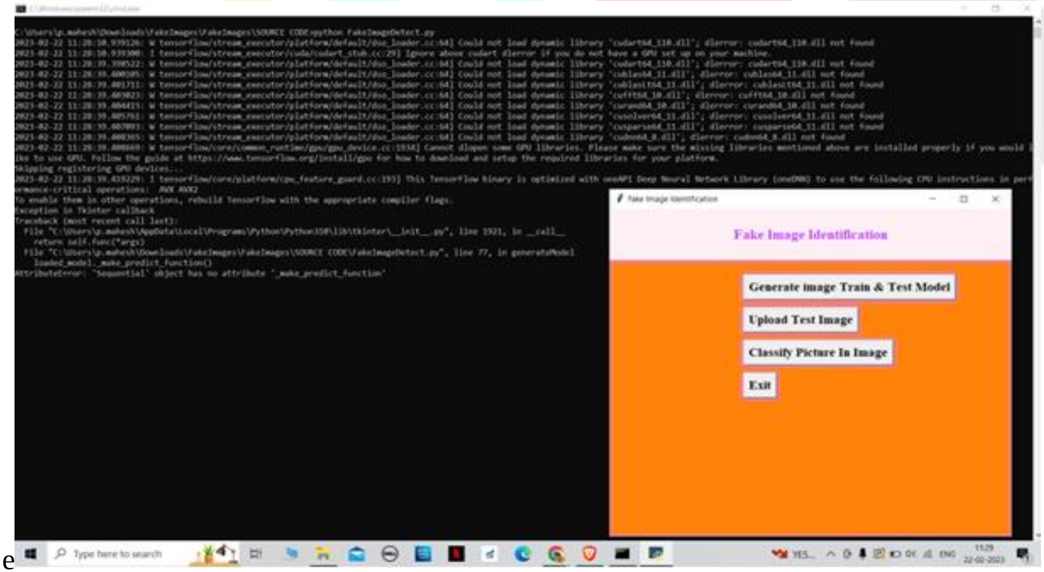
```
C:\Users\p.mahesh\Downloads\FakeImages\FakeImages\SOURCE CODE>python FakeImageDetect.py
2023-02-22 11:27:48.946448: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2023-02-22 11:27:48.946764: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
```

Screenshot 2: Initializing Of Code


C:\Windows\system32\cmd.exe
Fake Image Identification
Fake Image Identification
Generate image Train & Test Model
Upload Test Image
Classify Picture In Image
Exit
Type here to search
YES...
ENG
11:28
22-02-2023

```
C:\Users\p.mahesh\Downloads\FakeImages\FakeImages\SOURCE CODE>python FakeImageDetect.py
2023-02-22 11:28:10.939126: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2023-02-22 11:28:10.939300: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
```

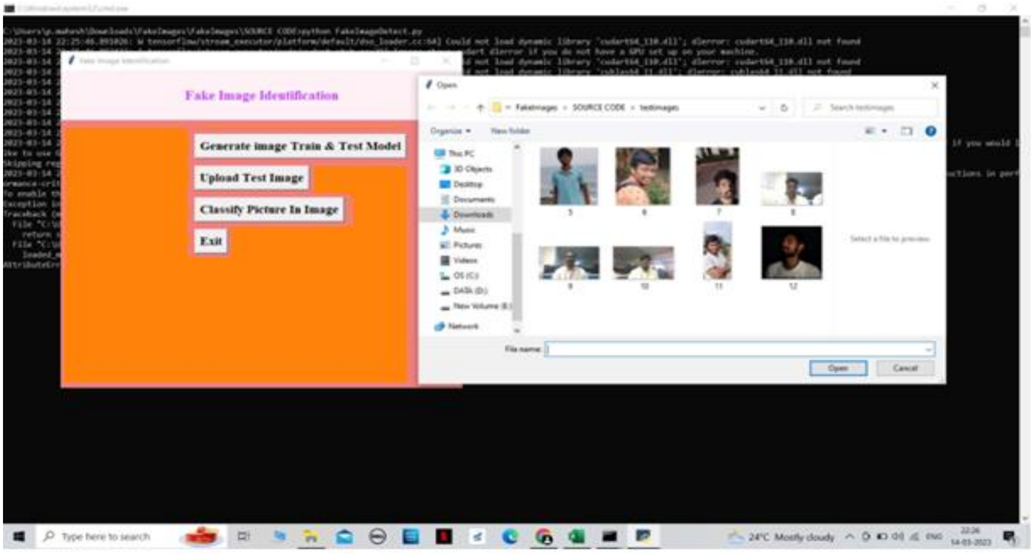
Screenshot 3: User Interface


C:\Windows\system32\cmd.exe
Fake Image Identification
Fake Image Identification
Generate image Train & Test Model
Upload Test Image
Classify Picture In Image
Exit
Type here to search
YES...
ENG
11:29
22-02-2023

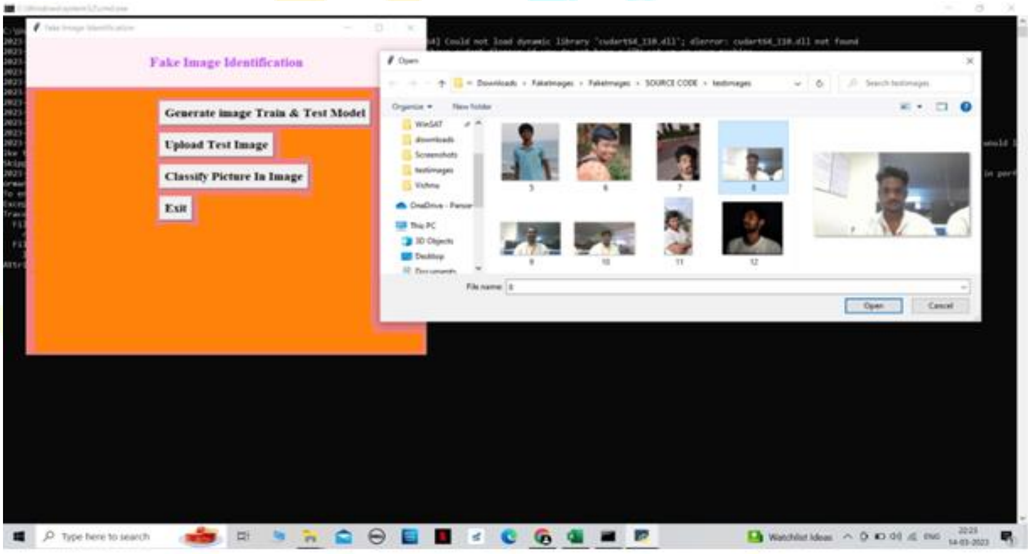
```
C:\Users\p.mahesh\Downloads\FakeImages\FakeImages\SOURCE CODE>python FakeImageDetect.py
2023-02-22 11:28:10.939126: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2023-02-22 11:28:10.939300: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2023-02-22 11:28:39.398522: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2023-02-22 11:28:39.400105: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublas64_11.dll'; dlerror: cublas64_11.dll not found
2023-02-22 11:28:39.401711: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublasLt64_11.dll'; dlerror: cublasLt64_11.dll not found
2023-02-22 11:28:39.403023: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cufft64_10.dll'; dlerror: cufft64_10.dll not found
2023-02-22 11:28:39.404415: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'curand64_10.dll'; dlerror: curand64_10.dll not found
2023-02-22 11:28:39.405761: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusolver64_11.dll'; dlerror: cusolver64_11.dll not found
2023-02-22 11:28:39.407093: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusparse64_11.dll'; dlerror: cusparse64_11.dll not found
2023-02-22 11:28:39.408365: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudnn64_8.dll'; dlerror: cudnn64_8.dll not found
2023-02-22 11:28:39.408669: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1934] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned above are installed properly if you would l
ike to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2023-02-22 11:28:39.419229: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in perf
ormance-critical operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\p.mahesh\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py", line 1921, in __call__
    return self.func(*args)
  File "C:\Users\p.mahesh\Downloads\FakeImages\FakeImages\SOURCE CODE\FakeImageDetect.py", line 77, in generateModel
    loaded_model._make_predict_function()
AttributeError: 'Sequential' object has no attribute '_make_predict_function'
```

e

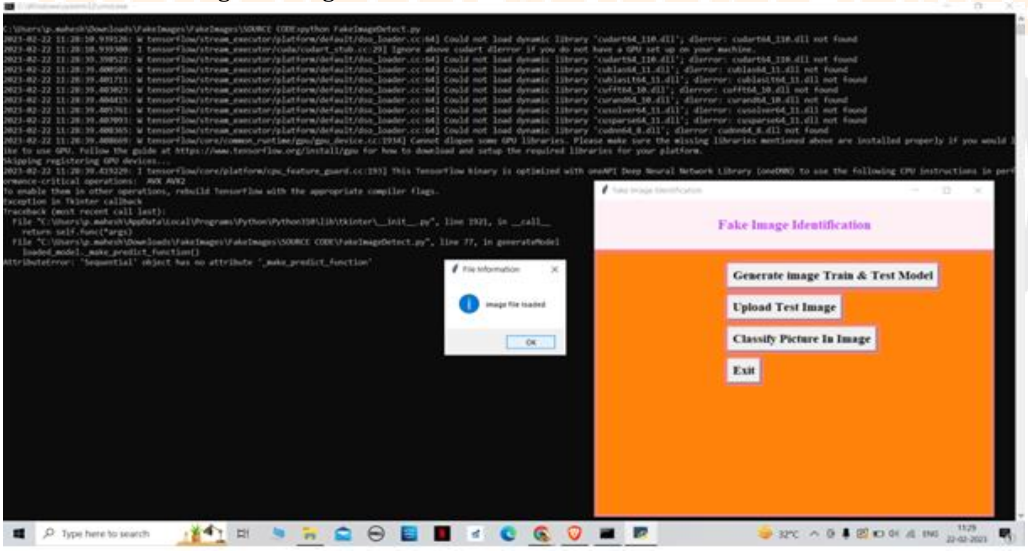
Screenshot 4: Then We Have To Generate Image Train And Test Model


Fake Image Identification
Generate image Train & Test Model
Upload Test Image
Classify Picture In Image
Exit
Open
Open
Cancel

Screenshot 5: the images with data base


Fake Image Identification
Generate image Train & Test Model
Upload Test Image
Classify Picture In Image
Exit
Open
Open
Cancel

Screenshot 6: selecting the image from the data base



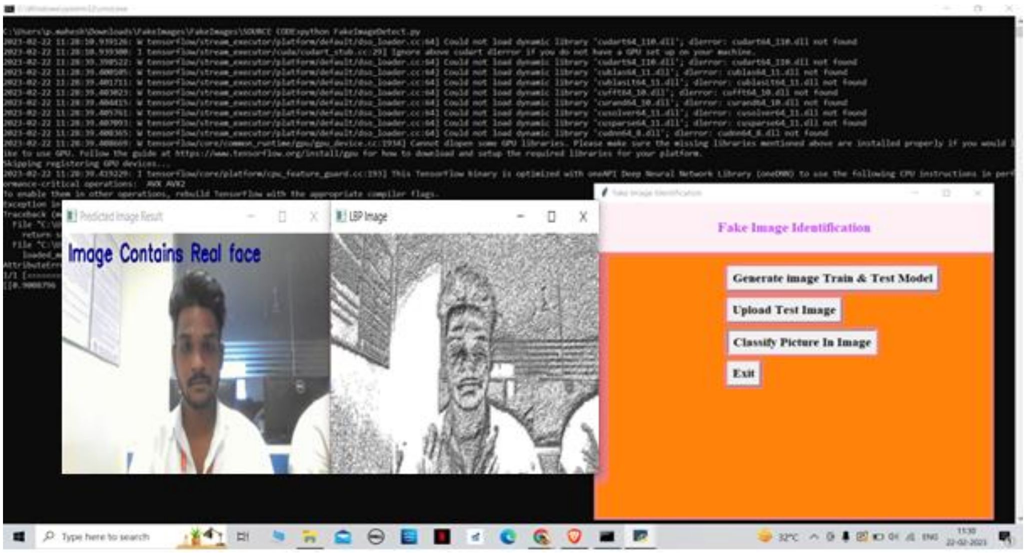
```
C:\Users\p.mahesh\Downloads\FakeImages\FakeImages\SOURCE CODE>python FakeImageDetect.py
2023-02-22 11:28:10.939126: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2023-02-22 11:28:10.939300: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2023-02-22 11:28:39.398522: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2023-02-22 11:28:39.400505: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublas64_11.dll'; dlerror: cublas64_11.dll not found
2023-02-22 11:28:39.401711: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublasLt64_11.dll'; dlerror: cublasLt64_11.dll not found
2023-02-22 11:28:39.403023: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cufft64_10.dll'; dlerror: cufft64_10.dll not found
2023-02-22 11:28:39.404415: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'curand64_10.dll'; dlerror: curand64_10.dll not found
2023-02-22 11:28:39.405761: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusolver64_11.dll'; dlerror: cusolver64_11.dll not found
2023-02-22 11:28:39.407093: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusparse64_11.dll'; dlerror: cusparse64_11.dll not found
2023-02-22 11:28:39.408365: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudnn64_8.dll'; dlerror: cudnn64_8.dll not found
2023-02-22 11:28:39.408669: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1934] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned above are installed properly if you would l
ike to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2023-02-22 11:28:39.419229: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in perf
ormance-critical operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\p.mahesh\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py", line 1921, in __call__
    return self.func(*args)
  File "C:\Users\p.mahesh\Downloads\FakeImages\FakeImages\SOURCE CODE\FakeImageDetect.py", line 77, in generateModel
    loaded_model._make_predict_function()
AttributeError: 'Sequential' object has no attribute '_make_predict_function'
```

File Information

image file loaded

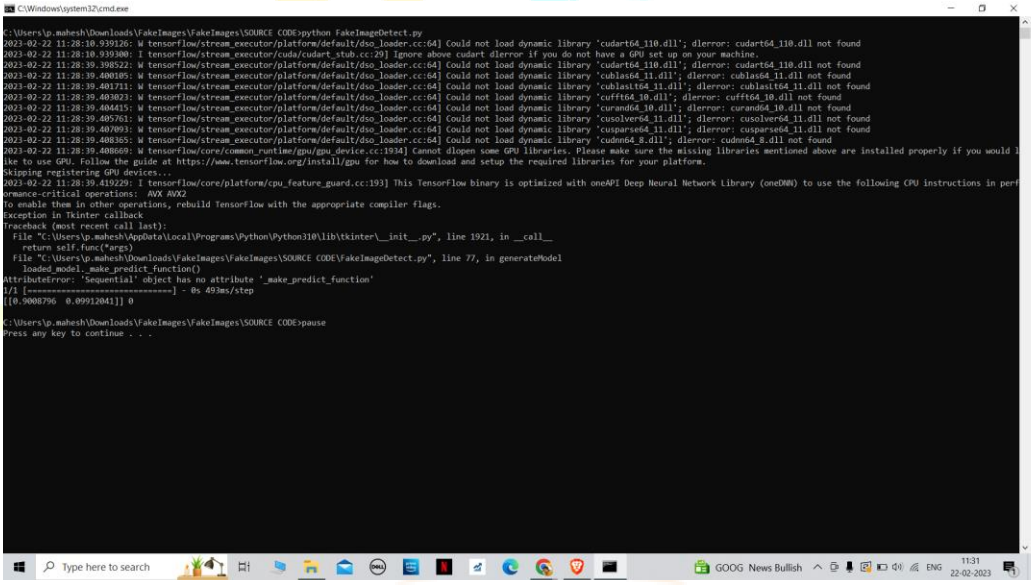
OK
Fake Image Identification
Generate image Train & Test Model
Upload Test Image
Classify Picture In Image
Exit

Screenshot 7: The selected image is uploaded



```
C:\Users\p.mahesh\Downloads\FakeImages\FakeImages\SOURCE CODE>python FakeImageDetect.py
2023-02-22 11:28:10.939126: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2023-02-22 11:28:10.939300: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2023-02-22 11:28:39.398522: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2023-02-22 11:28:39.400105: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublas64_11.dll'; dlerror: cublas64_11.dll not found
2023-02-22 11:28:39.401711: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublasLt64_11.dll'; dlerror: cublasLt64_11.dll not found
2023-02-22 11:28:39.403023: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cufft64_10.dll'; dlerror: cufft64_10.dll not found
2023-02-22 11:28:39.404415: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'curand64_10.dll'; dlerror: curand64_10.dll not found
2023-02-22 11:28:39.405761: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusolver64_11.dll'; dlerror: cusolver64_11.dll not found
2023-02-22 11:28:39.407093: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusparse64_11.dll'; dlerror: cusparse64_11.dll not found
2023-02-22 11:28:39.408365: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudnn64_8.dll'; dlerror: cudnn64_8.dll not found
2023-02-22 11:28:39.408669: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1934] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned above are installed properly if you would l
ike to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2023-02-22 11:28:39.419229: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in perf
ormance-critical operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Exception in
Traceback (m
  File "C:\Us
    return
  File "C:\U
    loaded_m
AttributeEr
1/1 [=======
[[0.9008796
```

Screenshot 8: The upload image is run behead of program and given the results



```
C:\Users\p.mahesh\Downloads\FakeImages\FakeImages\SOURCE CODE>python FakeImageDetect.py
2023-02-22 11:28:10.939126: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2023-02-22 11:28:10.939300: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2023-02-22 11:28:39.398522: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2023-02-22 11:28:39.400105: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublas64_11.dll'; dlerror: cublas64_11.dll not found
2023-02-22 11:28:39.401711: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublasLt64_11.dll'; dlerror: cublasLt64_11.dll not found
2023-02-22 11:28:39.403023: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cufft64_10.dll'; dlerror: cufft64_10.dll not found
2023-02-22 11:28:39.404415: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'curand64_10.dll'; dlerror: curand64_10.dll not found
2023-02-22 11:28:39.405761: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusolver64_11.dll'; dlerror: cusolver64_11.dll not found
2023-02-22 11:28:39.407093: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusparse64_11.dll'; dlerror: cusparse64_11.dll not found
2023-02-22 11:28:39.408365: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudnn64_8.dll'; dlerror: cudnn64_8.dll not found
2023-02-22 11:28:39.408669: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1934] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned above are installed properly if you would l
ike to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2023-02-22 11:28:39.419229: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in perf
ormance-critical operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\p.mahesh\AppData\Local\Programs\Python\Python310\lib\tkinter\__init__.py", line 1921, in __call__
    return self.func(*args)
  File "C:\Users\p.mahesh\Downloads\FakeImages\FakeImages\SOURCE CODE\FakeImageDetect.py", line 77, in generateModel
    loaded_model._make_predict_function()
AttributeError: 'Sequential' object has no attribute '_make_predict_function'
1/1 [==============================] - 0s 493ms/step
[[0.9008796  0.09912041]] 0

C:\Users\p.mahesh\Downloads\FakeImages\FakeImages\SOURCE CODE>pause
Press any key to continue . . .
```

Screenshot 9: The program is exited

6. CONCLUSION

In this study, we present a unique paired learning based common fake feature network to accurately identify fake face/general pictures produced by cutting-edge GANs By combining the cross-layer feature representations into the final fully connected layers, the proposed CFFN maybe utilised to train middle & high level & discriminative fake features. The performance of false picture detection may be further enhanced using the suggested paired learning.

The suggested fake image detector should be able to recognise the false picture produced by a fresh GAN with the help of the proposed pairwise learning. Our test findings showed that the suggested strategy works better in terms of precision and recall rate than other cutti ng-edge schemes.

7. ACKNOWLEDGEMENT

We thank CMR Technical Campus for supporting this paper titled "Uses of Deep Learning for Fake Image Detection", which provided good facilities and support to accomplish our work. Sincerely thank our Chairman, Director, Deans, Head Of the Department, Department Of Computer Science and Engineering, Guide and Teaching and Non- Teaching faculty members for giving valuable suggestions and guidance in every aspect of our work

8. REFERENCES

[1] Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. arXiv Preprint, arXiv:1710.10196 2017. 256.

[2] Brock, A.; Donahue, J.; Simonyan, K. Large scale gan training for high fidelity natural image synthesis. arXiv Preprint, arXiv:1809.11096 2018

[3] Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent 259 adversarial networks. arXiv Preprint, 2017.

[4] Hsu, C.; Lee, C.; Zhuang, Y. Learning to detect fake face images in the Wild. 2018 International Symposium 264 on Computer, Consumer and Control (IS3C), 2018, pp. 388–391. doi:10.1109/IS3C.2018.00104.

[5] H.T. Chang, C.C. Hsu, C.Y.a.D.S. Image authentication with tampering localization based on watermark 266 embedding in wavelet domain. Optical Engineering 2009, 48, 057002.

[6] Huaxiao Mo, B.C.; Luo, W. Fake Faces Identification via Convolutional Neural Network. Proc. of the ACM Workshop on Information Hiding and Multimedia Security. ACM, 2018, pp. 43–47.