Image Manipulation Detection through Machine Learning

K. Nagarjuna[1], G. Sirisha[2], A.G.L. Vyshnavi[3], V. Vamsi Krishna [4], T. Anil Kumar[5]

[1]*Assistant Professor, Department of CSE-Artificial Intelligence and Machine Learning, S.R.K Institute of Technology, NTR, Andhra Pradesh, India,* ***gandhamsiri2003@gmail.com***

[2]*student, Department of CSE-Artificial Intelligence and Machine Learning, S.R.K Institute ofTechnology, NTR, Andhra Pradesh, India*

[3]*student, Department of CSE-Artificial Intelligence and Machine Learning, S.R.K Institute ofTechnology, NTR, Andhra Pradesh, India*

[4]*student, Department of CSE-Artificial Intelligence and Machine Learning, S.R.K Institute ofTechnology, NTR, Andhra Pradesh, India*

[5]*student, Department of CSE-Artificial Intelligence and Machine Learning, S.R.K Institute ofTechnology, NTR, Andhra Pradesh, India*

Abstract— In present days fake images are becoming more realistic with high-quality, even hard to detect through human eyes. Biometric technology helps in recognizing a person's identity, but criminals alter their looks, and psychological behavior to deceive the recognizing system. As new types of fake images are emerging rapidly, developing new detection systems is becoming a challenging task. In this project, we explore this problem by using machine learning methodologies and image preprocessing. To overcome this problem, we are employing a method called Deep Texture Feature Extraction from images and then building a machine-learning model using the CNN algorithm. LBPNET, a machine learning convolution neural network, is the name of the network we created for this research to identify manipulated face photographs. Here, we will first extract LBP from the images, and then we will train the convolution neural network on the LBP descriptor images to produce the training model. Every time a new test image is uploaded, the training model will use that image to determine if the test image is manipulated or not.

Keywords— Image Preprocessing, LBP, Texture Feature Extraction, Machine Learning, LBPNet, CNN, Image Manipulation Detection.

I. INTRODUCTION

In the traditional image forgery detection approach, two types of forensics scheme are widely used: active schemes and passive schemes. With the active schemes, the externally additive signal (i.e., watermark) will be embedded in the source image without visual artifacts. To identify whether the image has been tampered or not, the watermark extraction process will be performed on the target image to restore the watermark. The extracted watermark image can be used to localize or detect the tampered regions in the target image. However, there is no "source image" for the generated images by GANs such that the active image forgery detector cannot be used to extract the watermark image. The second one-passive image forgery detector–uses the statistical information in the source image that will be highly consistent between different images. With this property, the intrinsic statistical information can be used to detect the fake region in the image. However, the passive image forgery detector cannot be used to identify the fake image generated by GANs since they are synthesized from the low-dimensional random vector. Nothing changes in the generated image by GANs because the fake image is not modified from its original image. The objective of this project is to identify manipulated images (Fake images).

II. LITERATURE SURVEY

According to a study conducted by Zheng et al. [2] (2018), the identification of fake news and images is very difficult, as fact-finding of news on a pure basis remains an open problem and

few existing models Can be used to resolve the problem. It has been proposed to study the problem of "detecting false news." Through a thorough investigation of counterfeit news, many useful properties are determined from text words and pictures used in counterfeit news. There are some hidden characteristics in words and images used in fake news, which can be identified through a collection of hidden properties derived from this model through various layers. A pattern called TI-CNN has been proposed. By displaying clear and embedded features in a unified space, TI-CNN is trained with both text and image information at the same time.

Raturi's 2018 architecture [3] was proposed to identify counterfeit accounts in social networks, especially on Facebook. In this research, a machine learning feature was used to better predict fake accounts, based on their posts and the placement on their social networking walls. Support Vector Machine (SVM) and Complement Naïve Bayes (CNB) were used in this process, to validate content based on text classification and data analysis. The analysis of the data focused on the collection of offensive words, and the number of times they were repeated. For Facebook, SVM shows a 97% resolution where CNB shows 95% accuracy in recognizing Bag of Words (BOW) -based counterfeit accounts. The results of the study confirmed that the main problem related to the safety of social networks is that data is not properly validated before publishing.

Aphiwongsophon and Chongstitvatana [5], aimed to use automated learning techniques to detect counterfeit news. Three common techniques were used in the experiments: Naïve Bayes, Neural Network, and Support Vector Machine (SVM). The normalization method is a major step to disinfect data before using the automatic learning method to sort information. The results show Naïve Bayes to have a 96.08% accuracy in detecting counterfeit news. There are two other advanced methods, the Neural Network Machine and the Support Network (SVM), which achieve 99.90% accuracy.

Hsu, C ; Lee C et al.[12] have proposed a novel deep forgery discriminator(DeepFD) to detect fake images generated by state-of-the-art GANS based on contrastive loss. To address the existing shortcomings they had adopted contrastive lossin extracting the typical features of fake/generated images generated by different GANS.Their results have shown that their proposed system DeepFD had detected 94.7% fake images which are generated byseveral state of the art GANS.

A neural network was successfully trained in [6] by Kuruvilla et al. by comparing the error levels of 4000 real photos and 4000 fake images. With an impressive 83% success rate, the trained neural network was able to determine whether the image was real or phony. According to the findings, the spreading of false photos on social networks is dramatically decreased when using this software on mobile platforms. Furthermore, this can be applied as a fake image verification technique in digital authentication, the evaluation of court evidence, etc. By merging the output of the neural network (80%) with the findings of the metadata analysis (80%), it creates and tests a reliable false picture detecting algorithm.

III. EXISTING SYSTEM

In the existing system, passive techniques were used to identify the alterations that an image has undergone for its forgery. while capturing an image, additional hidden information is associated with it for authentication and forgery protection. The problem with existing fake image detection systems is that they can be used to detect only specific tampering methods like splicing, coloring, etc. We approached the problem using machine learning and neural networks to detect almost all kinds of tampering on images. The passive technique does not rely on additional metadata, but it only analyzes the image itself to uncover anomalies or alterations in the image itself. Copy-move means copying a part of an image and pasting it into another place of the same picture whereas splicing is about taking a part of an image and pasting it into

another.

Disadvantages:

- Complexity in analyzing the data.
- Prediction is a challenging task working on the model
- Coding is complex maintaining multiple methods.
- The library's support was not that much familiar.

IV. PROPOSED SYSTEM

In this project, we are planning to use LBP primarily based on machine learning Convolution Neural Network known as LBPNET to sight pretend face images. Here first we'll extract LBP from images. Then we will train LBP descriptor pictures with Convolution Neural Network to get a training model. Whenever we want to test a new image, we upload it in the application. It compares the extracted features pattern with the trained model and predicts whether the image contains a real or manipulated image. Because of its discriminative power and machine simplicity, the LBP texture operator has become a preferred approach in numerous applications such as object recognition and face recognition, since it is robust to changes in lighting and can be computed quickly. It is often seen as a unifying approach to the historically divergent applied mathematics and structural models of texture analysis.

Advantages:

- It will extract the hidden features from the images.
- It gives better and more accurate results and minimum time requirement.
- Various libraries help to analyze the data.
- Results will be accurate compared to other methodologies.

a) Methodology

We chose the LBPNet Algorithm to extract texture features from images and then train the model using convolutional neural networks. The Proposed system is an application means to classify whether the image is fake or real using machine learning techniques and neural networks.

In this Project, we use the NUAA Photograph Imposter (fake) Database with images obtained from real and fake faces. We also used images and converted those images into LBP format. The database was created by Nanjing University of Aeronautics and Astronautics, China by conducting different sessions with people to capture their real faces and imposter faces. Below are some images from the LBP folder.

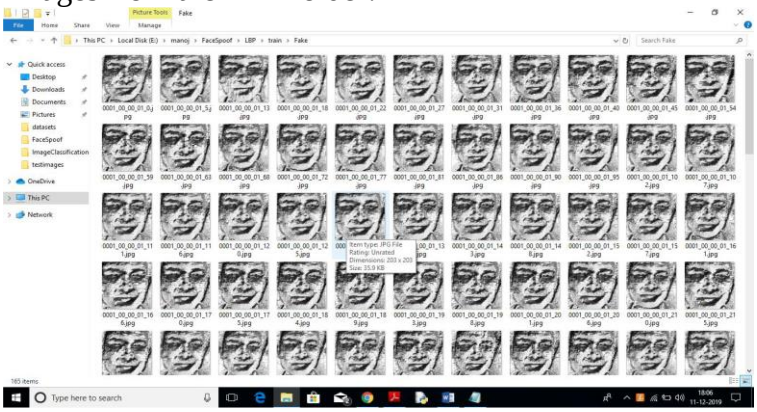

Fig-1: Dataset Image

The system architecture of the project helps to understand the flow of the application. It describes the step-by-step procedure of the complete project. Here, firstly we will import the dataset which is a process of loading and reading data from various resources. Then, the data will be preprocessed. In other words, the features of the data can now be easily interpreted by

the algorithm. Next, we will read all LBP images from the LBP folder and then train the CNN model with all those images to generate a training model. Once the training model has generated we will upload test image from 'test images'. The application will read this image and then extract Deep texture features from this image using the LBP algorithm. Finally, we need to apply a test image on the CNN train model to predict whether the test image contains a real image or a manipulated image.

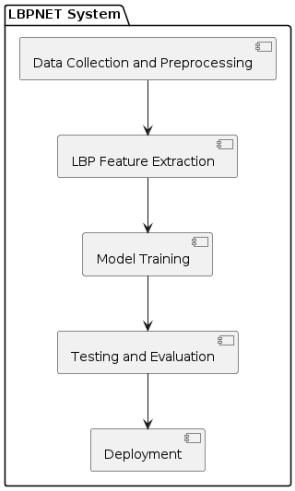

Fig-2: Proposed System Architecture

b) LBPNet Algorithm

Local binary patterns (LBP) are a type of visual descriptor used for classification in computer vision. It compares each pixel in an image to its surrounding pixels and encodes the results in a binary pattern. These binary patterns can then be used to classify the texture of the image or to compare the texture of different images.

The LBP feature vector, in its simplest form, is created in the following manner:

- Divide the examined window into cells (e.g. 3 X 3 pixels for each cell).
- For each pixel in a cell, compare the pixel to each of its 8 neighbors (on its left-top, left-middle, left-bottom, right-top, etc.). Follow the pixels along a circle, i.e. clockwise or counter-clockwise. Where the center pixel's value is greater than the neighbor's value, write "0". Otherwise, write "1".
- This gives an 8-digit binary number.
- Compute the histogram, over the cell, of the frequency of each "number" occurring. This histogram can be seen as a 256-dimensional feature vector.
- Optionally normalize the histogram. Concatenate (normalized) histograms of all cells. This gives a feature vector for the entire window.

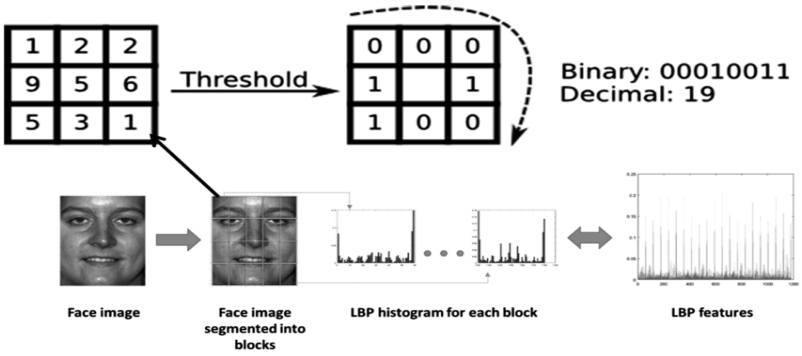

Fig-3: Representation of LBP features from Images

The feature vector can now be processed using the Support vector machine, extreme learning machines, or other machine learning algorithms to classify images. Such classifiers can be used for face recognition or texture analysis.

V. IMPLEMENTATION

A) Import or Load Dataset: Initially we collect the dataset i.e., images of fake and real faces of persons, and combine them as a data store which will be in the form of folders.

B) Data Preprocessing: We make use of the input images through file directories and preprocessing will be applied to the images for cleaning and preparing the data for the model. The preprocessing tasks include resizing the images, noise removal and normalization of data.

C) Generate Train & Test Model: In this module, we will read all LBP images from the LBP folder and then train the CNN model with all those images.

D) Upload Test Image: Here, we will upload test images from the 'test images' folder. The application will read this image and then extract Deep texture features from this image using the LBP algorithm.

E) Classify Picture In Image: This module applies test image on the CNN train model to predict whether the test image contains a real image or a manipulated fake image.

F) Deployment: Here we display the prediction results to the users through an interactive user interface.

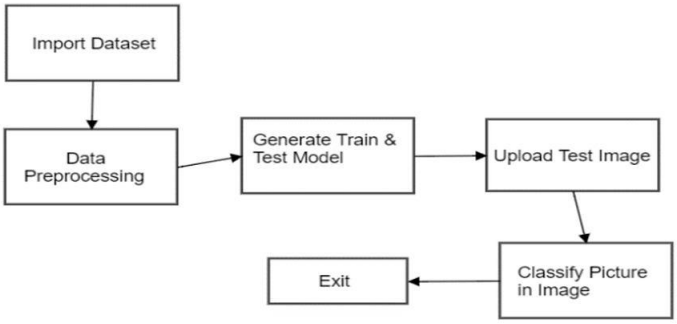

Fig-4: Implementation of proposed work

VI. SAMPLE SCREENSHOTS

To run this project click on the 'run.bat' file to get the below screen.

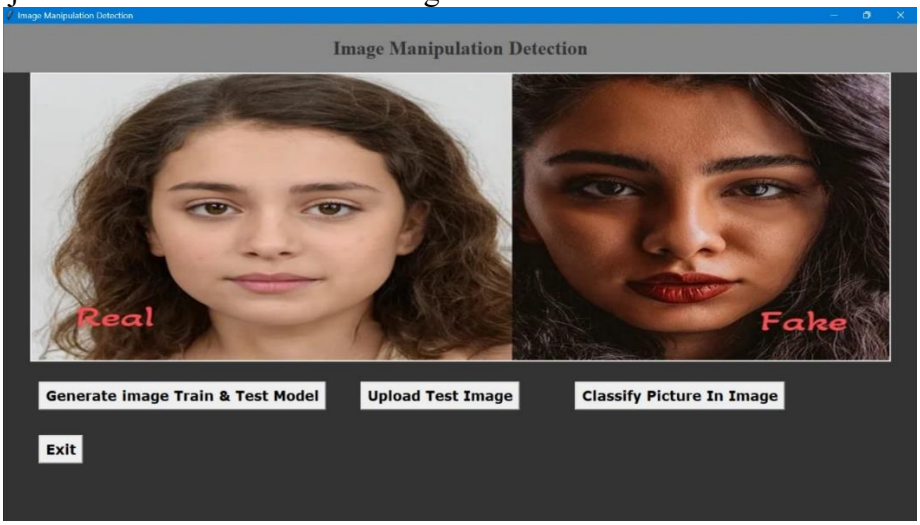

Fig-5: Output Screen 1

In above screen click on 'Generate Image Train & Test Model' button to generate CNN model using LBP images contains inside LBP folder.

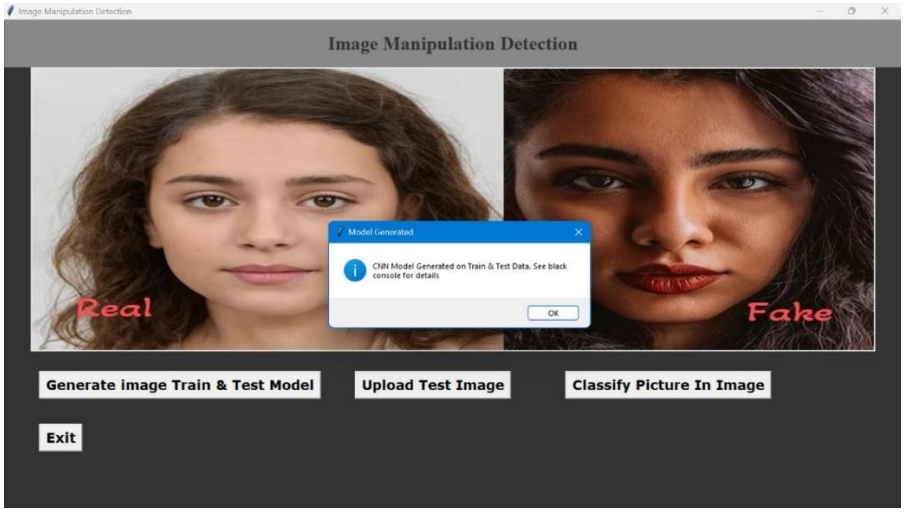

Fig-6: Output Screen 2

In above screen we can see CNN LBPNET model generated. Now click on 'Upload Test Image' button to upload test image.

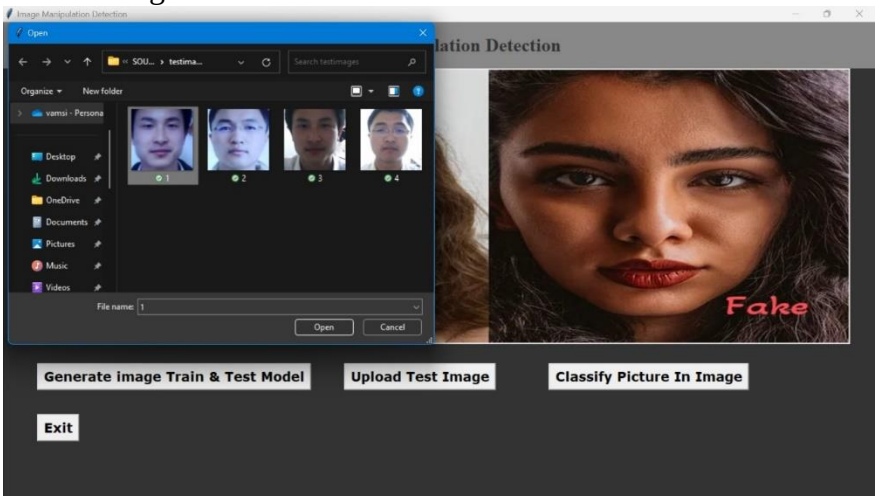

Fig-7 : Output Screen 3

In above screen we can see two faces are there from same person but in different appearances. In above screen, we can see all real face will have normal light and in fake faces people will try some editing to avoid detection but this application will detect whether face is real or fake.

In the below screen we have uploaded 1.jpg and after upload click on open button to get the image file uploaded.

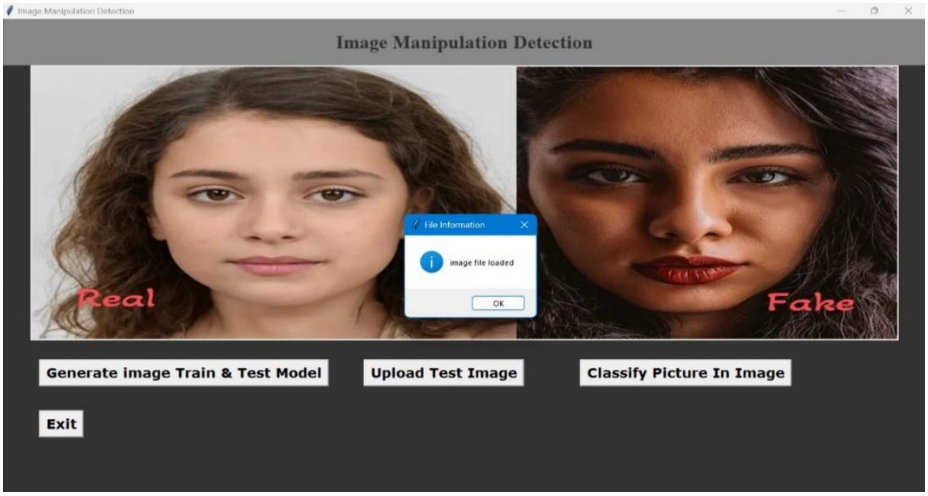

Fig-8: Output Screen 4

And now click on 'classify Picture in Image' to get below results.

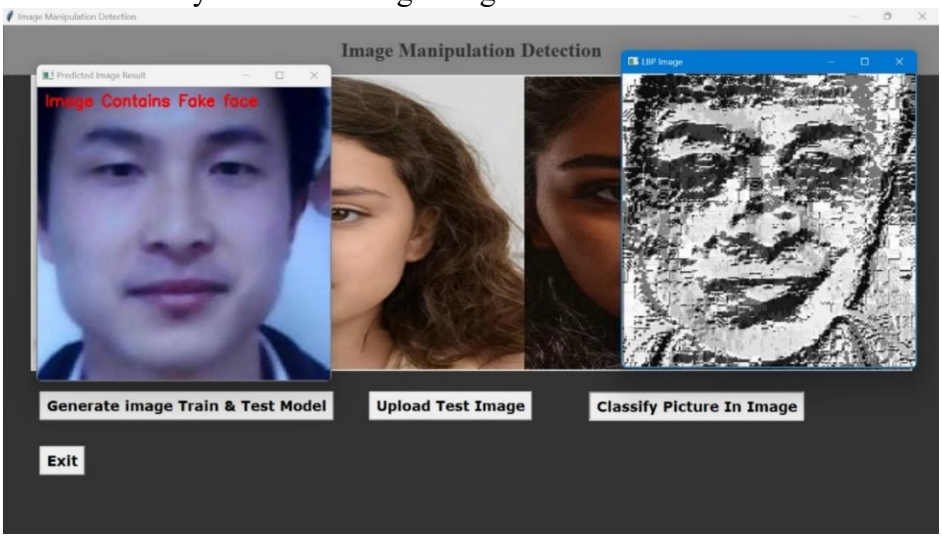

Fig-9 : Output Screen 5

In above screen we are getting result as image contains Fake face. As with these images, you can also try on some real-world images.

VII. CONCLUSION

In this paper, we have proposed an LBP-based neural network based on pairwise learning, to detect the manipulated /general images successfully. The proposed LBPNet can be used to learn the middle-level and high-level discriminative fake features by aggregating the cross-layer feature representations into the last fully connected layers. Neural networks used in this project are trained with LBP descriptor images and the model was designed according to machine learning techniques. The proposed model has shown a result of 96.7% accuracy and a precision rate of 95.4%. Our experimental results demonstrated that the proposed method outperforms other state-of-the-art schemes in terms of accuracy, precision, and recall rate.

VIII. FUTURE SCOPE

For future work, we can suggest using a more complex and deeper model for unpredictable problems. Integration of deep neural networks with the theory of enhanced learning, where the model is more effective. Neural network solutions rarely take into account non-linear interactions and non-monotonous short-term sequential patterns, which are necessary to model user behaviour in sparse sequence data. A model may be integrated with neural networks to solve this problem. The dataset could be increased and another type of image could be used for training, for example, gray-scale images. Like a neural network, CNN and its variants can also be optimized for large datasets, which is often the case when classifying objects and images.

REFERENCES

[1]. K. Ravi, (2018). Detecting fake images with Machine Learning. Harkuch Journal
[2]. L. Zheng, Y. Yang, J. Zhang, Q. Cui, X. Zhang, Z. Li, et al. (2018). TICNN: Convolutional Neural Networks for Fake News Detection. United States
[3]. R. Raturi, (2018). Machine Learning Implementation for Identifying Fake Accounts in Social Networks. International Journal of Pure and Applied Mathematics, 118(20), 4785-4797.
[4] .J. Bunk, J. Bappy, H. Mohammed, T. M. Nataraj, L., Flenner, A., Manjunath, B., et al. (2017). Detection and Localization of Image Forgeries using Resampling Features and Deep Learning. The University of California, Department of Electrical and Computer Engineering, USA.

[5]. S. Aphiwongsophon, & P. Chongstitvatana, (2017). Detecting Fake News with Machine Learning Method. Chulalongkorn University, Department of Computer Engineering, Bangkok, Thailand.
[6]. M. Villan, A. Kuruvilla, K. J. Paul, & E. P. Elias, (2017). Fake Image Detection Using Machine Learning. IRACST—International Journal of Computer Science and Information Technology & Security (IJCSITS).
[7]. S. Shalev-Shwartz, & S. Ben-David, (2014). Understanding Machine Learning: From Theory to Algorithms. New York: Cambridge University Press.
[8]. 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
[9]. Brock, A.; Donahue, J.; Simonyan, K. Large scale gan training for high fidelity natural image synthesis. arXiv Preprint, arXiv:1809.11096 2018.
[10]. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent 259 adversarial networks. arXiv Preprint, 2017.
[11]. AI can now create fake porn, making revenge porn even more complicated,. http://theconversation.com/ai-can-now-create-fake-porn-making-revenge-porn-even-more-complicated-92267, 262 2018.
[12]. 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.