DeepFake and Steganography in an Image Using Deep Learning

Prof. Pramode Bide
Department of Computer Engineering, Mumbai, MH , India.
pramod_bide@spit.ac.in

Rahul Tandel
Department of Computer Engineering, Mumbai, MH, India.
rahul.tandel@spit.ac.in

Omkar Padir
Department of Computer Engineering, Mumbai, MH, India.
omkar.padir@spit.ac.in

Sumit Thakare
Department of Computer Engineering, Mumbai, MH, India.
sumit.thakare@spit,ac,in

Department of computer Engineering , Sardar Patel Institute of Technology, Mumbai, India

Abstract— This paper focuses on a process of hiding data to image by using the least significant bit (LSB) and the symmetric key between the sender and the receiver. Here we have to choose the bits that will get the minimum resolution between the original image and stego image. This paper introduces a best approach for Least Significant Bit (LSB) based on image steganography that enhances the existing LSB substitution techniques to improve the security level of hidden information. It is a new approach to substitute LSB for RGB true color image. This paper further explains how the encryption and decryption processes are done.. This paper provides a way to automatically and effectively detect facial interference in videos, and focuses mainly on the latest two techniques used to produce fake videos. conditions and to demonstrate that the proposed metric learning method can be very effective in making such a category.

Keywords— Image Classification, Deepfakes, Image Manipulation, Steganography

INTRODUCTION

Image Steganography allows for two parties to communicate secretly and covertly. Steganography is a technique to hide information from the observer to establish an invisible communication [1]. Generally a steganographic system consists of cover media into which the secret information is embedded. The embedding process produces a stego medium by replacing the information with data from hidden messages. To hide hidden information, steganography gives a large opportunity in such a way that someone cannot know the presence of the hidden message. The goal of modem steganography is to keep its information undetectable. Generally secret information is stored into the specific position of Least Significant Bit (LSB) of a cover image which is the carrier to embed messages [1, 2, 3, 4]. Anyone can ensure that the specific position of LSB contains secret information. So it is easy to recover the secret information for anyone by using a retrieval method. The main intention of image steganography is to ensure security of hidden information. For security purposes, we have introduced a new approach of LSB based image steganography. Here we are adding a secret key which ensures the security of hidden information. The insertion of hidden information is totally controlled by the secret key[4]. This secret key decides the appropriate position of hidden information. It is very difficult to retrieve the hidden information without the same secret key. So by using a secret key, we can increase the security level of the hidden information in LSB based image steganography.

With the rapid growth of online streaming platforms, there is a great need to check video authenticity. The rise of deepfakes [9] in recent years raises serious concerns about the authenticity of digital content by the media and other online forums. Productive structures are excellent at helping to maximize the performance of in-depth learning structures by satisfying the need for large data sets, and often exploring the creative potential of in-depth learning. However, methods such as these also led to the Deepfakes, now being used for evil purposes to deceive images of politicians, famous actors, etc. Many politicians and actors became victims of the Deepfakes. For criminal purposes, forensic videos are replaced using new methods such as face change and face change. Various apps use a person's face to transform into fun and complex images such as age change, gender change, etc. In exchange, users provide facial data to these companies, which may be used for malicious purposes. When deceptive videos are shared in public applications, their quality decreases to make it easier to upload. The following two lines are examples of real sequences of relevant databases. and downloading those apps. In high quality videos, a small amount of misunderstanding around the face can be seen. However, in low quality videos, users can not distinguish between videos that are real or fake and the videos are transmitted[9] to large groups of people. Such deception can have far-reaching effects, from politics to the entertainment industry. Opposition (sight) as video deception, a few algorithms using hand-made features, in-depth reading algorithms are tested.

Steganography

Image Steganography is the process of hiding information which can be text, image or video within a cover image. Confidential information is hidden in such a way that it is invisible to the public eye. In-depth learning technology, which has already emerged as a powerful tool in a variety of programs including image steganography, has received more attention recently.

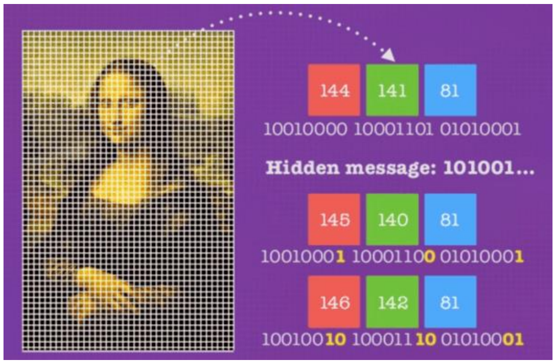


Fig 1.1 Stego hidden message in image.

Hidden methods have been around for a long time but their value has only increased recently[5]. As shown on Fig 1.1 the main reason for the increase in online data traffic and social media. Although the objectives of cryptography and steganography are similar, there are subtle differences. Cryptography makes data unbreakable and unreadable but the cipher text is visible to human eyes. Steganography, which is used to hide information publicly, allows the use of a wide variety of confidential information such as image, text, audio, video and files. Digital tagging is another way in which confidential information is embedded in order to claim ownership. Cryptography is a popular method used to hide information, however, steganography is gaining popularity in recent times.

Image of Steganography: A Review of Recent Developments multimedia data such as image, text, file or video [6]. Picture steganography is a way of hiding an image in the middle of another picture. In steganography, the cover image is manipulated in such a way that the hidden data does not look like that which makes it less suspicious as in the case of cryptography. In contrast, Steganalysis is used to determine the presence of any a secret message covered in the image and a hidden output data

The main purpose of this paper is to review existing methods, present trends and discuss current challenges in the courses. In line with these studies, data sets are publicly available and widely used[7], and thought-out metrics are also discussed. Finally, to compare performance between methods and possible discussion that identifies gaps in current studies, the pros and cons of methods are explained.

A. STEGANOGRAPHY PROCESS:

Secret Message: The data that you need to insert inside the digital media.. Stego-key: The key used in the Steganography process[7]. Cover Media: The medium utilized in Steganography procedure, for example, picture, video and audio. Sender Algorithm: The technique utilized in this Steganography process. Stego-Media: The media coming about because of including the mystery message into a spread media utilizing Stego-key and encoding calculation. As shown in Fig 1.2. Receiver Algorithm: The technique used to extract the mystery message from media utilizing Stego-key.

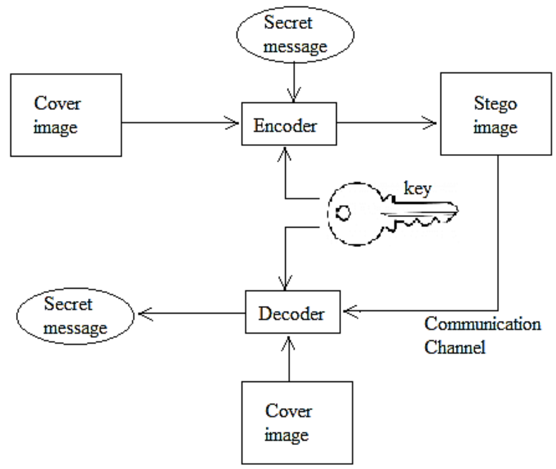


Fig 1.2 Steganography block diagram

B. STEGANOGRAPHY USING LSB

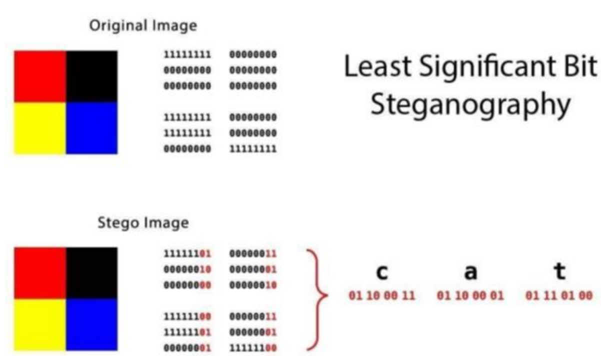


Fig 1.3 How message is Hidden using LSB.

The well-known strategy that is utilized for steganography is the LSB. And additionally the prominent technique for present day, steganography is to utilize LSB of picture‘s pixel data Fig 1.3 . This investigation is utilized for one piece of the LSB[5]. It inserts each piece of the double content piece with one piece of every pixel in the first picture. This strategy works when the record is longer than the message document and if picture is grayscale, when applying LSB strategies to every byte of a 24 bit picture, three bits can be encoded into every pixel[7][8] Example: We can use images to hide things if we replace the last bit of every color's byte with a bit from the message.

Image with 3 pixel

Message A-01000001

Image with 3 pixels

Pixel 1:	11111000	11001001	00000011
Pixel 2:	11111000	11001001	00000011
Pixel 3:	11111000	11001001	00000011

Fig. 2 *Message A before encryption*

Now we hide our message in the image.
Message A- 01000001

Message A- 01000001

Pixel 1:	11111000	11001001	00000010
Pixel 2:	111110000	11001000	00000010
Pixel 3:	111110000	11001001	00000011

Fig. 3 Message *A after encryption*

The simplest approach to hiding data within an image is called least significant bit (LSB) insertion. For 24-bit true color image, the amount of changes will be minimal and indiscernible to the human eye[7]. As an example, suppose that we have three adjacent pixels (nine bytes) with the following RGB encoding:

10010101	00001101	11001001
10010110	00001111	11001010
10011111	00010000	11001011

Now suppose we want to hide the following 9 bits of data **101101101**. If we overlay these 9 bits over the LSB of the 9 bytes above, we get the following (where bits in bold have been changed) pixels:

10010101	0000110**0**	11001001
1001011**1**	0000111**0**	1100101**1**
10011111	00010000	11001011

The following formula provides a very generic description of the pieces of the steganographic process:

cover image + hidden information = stego image

C. Hiding Technique of Hidden Information.

The secret key as in fig2 and fig3 encryption is converted into 10 array of bit stream. Secret key and Red matrix are used only for decision making to replace hidden information into either Green matrix or Blue matrix.

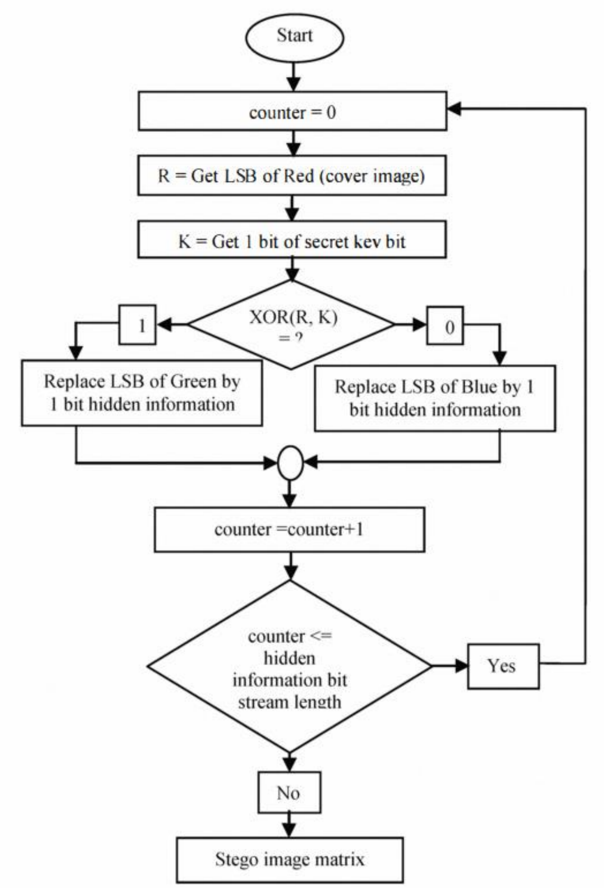


Fig 4 . Flow Chart to hide hidden infonnation into cover image

Each bit of secret key is XOR with each LSB of Red matrix. The resulting XOR value decides that the 1 bit of hidden information will be placed with either LSB[6][7] of Green matrix or Blue matrix. The same process will be continued until the hidden information is finished. The flow chart to hide hidden information into cover image is shown in Fig.4.

At Fig. 5, the LSB of Red matrix of pixel 1 is 0 and the first bit of secret key is 1. The XOR value of 0 and 1 is 1. In our method, if the XOR value is 1 then the LSB of Green matrix is replaced by the first bit of hidden information. If the XOR value is 0 then the LSB of Red matrix is replaced by the first bit of hidden information[8]. The 10 array of secret key is circular. The substitution process will be continued depending on the length of hidden information's 10 array.

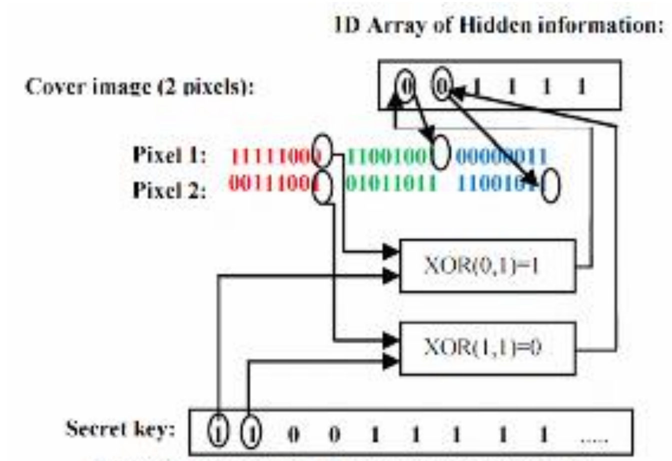


Fig. 5 1D array representation of hidden information

5. ENCODING

The advancements will clarify the encoding procedure of the (LSB+SPACING) calculation utilizing a Graphic User Interface (GUI) reproduction program:

Press on the File (open picture) to choose the Angry Bird picture from the drive.

Go to Steganography tab and press on Encode push catch (open content) to choose the mystery message from the local drive that is spared as a .txt record or type the message u want to encode. The confidential message that we need to execute it on this process is appeared in the figure, the content of the confidential message will likewise show up in the content on the program window.

At the finish of the Steganography procedure, choose the key[8][6] .The key that has been produced by the information of the confidential message and the Angry Bird picture utilizing the (LSB+SPACING) strategy is appeared in the figure 8.

. After that, the client can push on the Submit button and can save the image in its Drive. (Save Stego Image) to spare the stego picture on the plate refere fig 6.

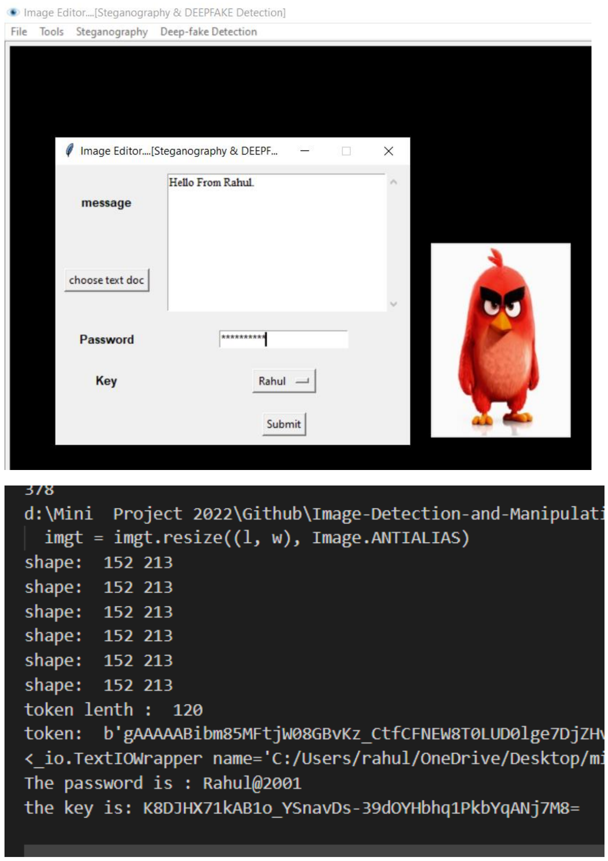


Fig 6. Our GUI and data encryption for hiding the message into image using key and password.

DECODING:

This strategy will be at the recipient, it expels the confidential message from the stego[2] picture subject to the common key between the sender and the receiver. Going with advancements we will show the decoding technique using the program that is showed up in the figure :

Press on the Steganography -> Decode button (Open Stego Image) to pick the stego picture from the Drive.

Now the client must provide proper password and the Key and by doing soo. The unwinding system will start to remove the confidential message from the stego picture.

By providing right Key and Password the confidential message will be displayed on the window. After that, the removed confidential message will appear in the substance region on the program window.

Finally, the customer can save the puzzle message that is appeared in substance on the program window by tapping on the push get (Save Text) to save it in the drive (.txt).As shown in fig 7

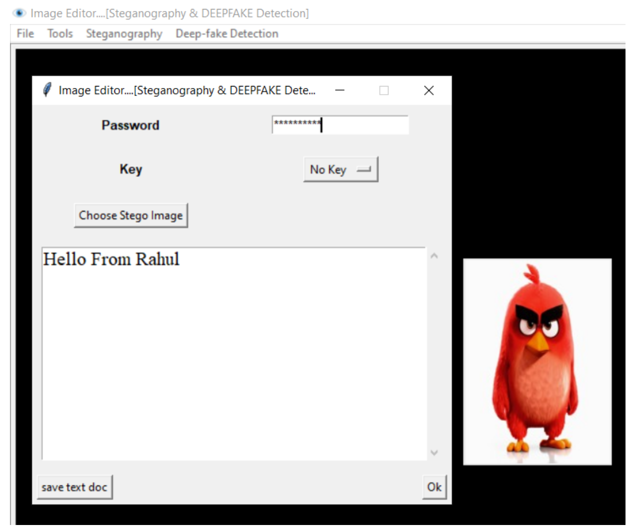


Fig 7.Data extraction form stego image using key and password.

DEEPFAKE:

Deepfake is a method that aims to remove the target person's face from another person's face in the video. The main idea is in the same training of the two autoencoders. Their properties can vary depending on the size of the outlet, the required training time, the expected level and the resources available. Traditionally, the default encoder determines the network coder network and the output network. The purpose of the codec is to reduce the size of coding data from the input layer to the reduced value of the variance. The goal of the decoder is to use the variable to remove the actual input limit. The development phase is done by comparing inputs and outputs produced and punishing the differences between the two approaches. To counter (detect) such as video manipulation, several algorithms using handcrafted features, deep learning algorithms, and lately GAN-based[10] methods are being explored. For instance, handcrafted approaches involve methods for steganalysis, detecting 3D head pose inconsistencies, etc. Several such existing approaches are summarized in [10] and [11]. However, there is still scope of improvement over the state-of-the-art arXiv:2003.08645v1 [cs.CV] 19 Mar 2020 for detecting deepfakes, especially on challenging data such as the Face Forensics (FF++) dataset [11].

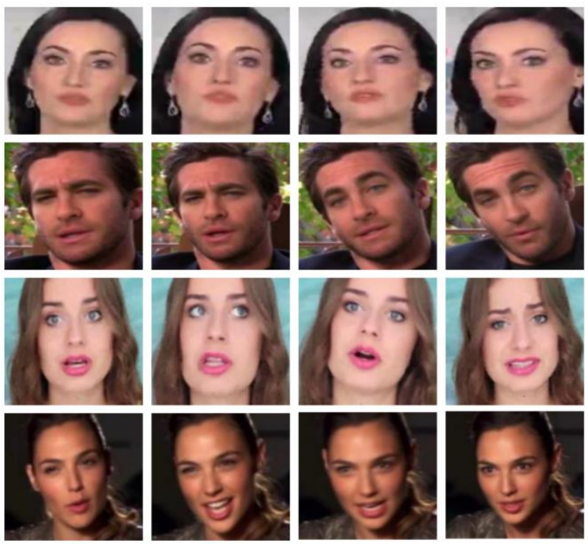


Fig.8 The first two rows depicts the deepfake frames of FF++ and CelebDF dataset. Next two rows are examples of original sequences of the respective datasets.

MTCNN

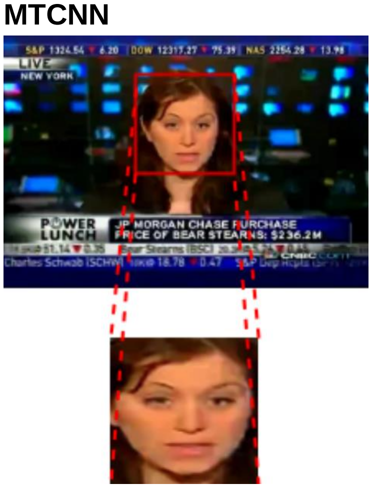


Fig 9. Extraction of face from frame using MTCNN algorithm.

Crops out images using Proposal, Refine and Output Net. Proposal network detect faces across multiple resolutions, then, refine net suppress the overlapping boxes using nonmax suppression, which is available online.

A. Proposed method

This section introduces a few practical ways to deal with Deepfake or Face2Face. It turns out that these two problems cannot be successfully solved with a unique network. However, due to the same nature of lying, the same network structures for both problems can produce good results. We are proposing the discovery of fake facial videos by setting our way to the mesoscopic level of analysis. Indeed, small-scale image-based analysis cannot be applied to compressed video context when image volume is significantly reduced. Similarly, at a high semantic level, the human eye finds it difficult to distinguish shaped images [12], especially when the image reflects the human face [13, 14]. This is why we propose to use the middle method using a deep neural network with a small number of layers.

The following architecture have achieved the best classification scores among all our tests, with a low level of representation and a surprisingly low number of parameters.It is based on well-performing networks for image classification [15, 16] that alternate layers of convolutions and pooling for feature extraction and a dense network for classification. Their source code is available online1 .

https://github.com/DariusAf/MesoNet.

Meso-4:

This network begins with a sequence of four layers of successive convolutions and pooling, and is followed by a dense network with one hidden layer. To improve generalization, the convolutional layers use ReLU activation functions that introduce non-linearities and Batch Normalization [17] to regularize their output and prevent the vanishing gradient effect, and the fully-connected layers use Dropout [15] to regularize and improve their robustness.

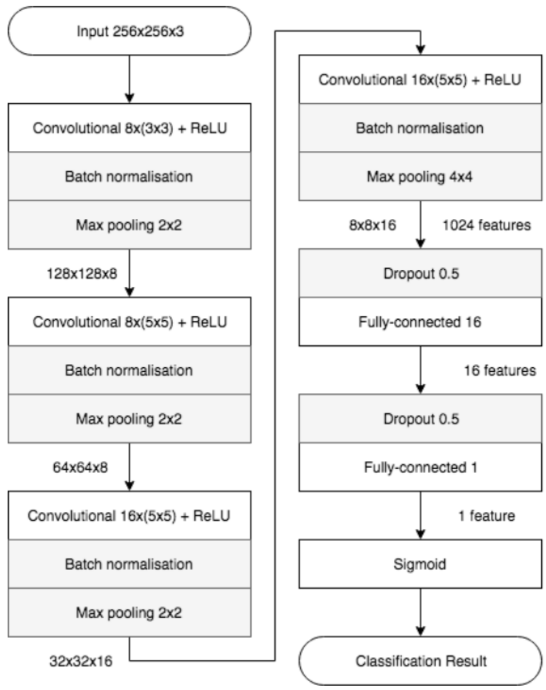


Fig.10 The network architecture of Meso-4. Layers and parameters are displayed in the boxes, output sizes next to the arrows.

In total, there are 27,977 trainable parameters for this network. Further details can be found on Figure 4.

3.1. Datasets

3.1.1 Deepfake dataset

To our knowledge, no dataset gathers videos generated by the Deepfake technique, so we have created our own.

Training auto-encoders for the forgery task requires several days of training with conventional processors to achieve realistic results and can only be done for two specific faces at a time. To have a sufficient variety of faces, we have rather chosen to download the profusion of videos available to the general public on the internet. Thus, 175 rushes of forged videos have been collected from different platforms. Their duration ranges from two seconds to three minutes and have a minimum resolution of 854 × 480 pixels. All videos are compressed using the H.264 codec but with different compression levels, which puts us in real conditions of analysis. An accurate study on the effect of compression levels is conducted on another dataset introduced in Section 3.1.2. All the faces have been extracted using the Viola-Jones detector [16] and aligned using a trained neural network for facial landmark detection [12]. In order to balance the distribution of faces, the number of selected frames for extraction per video is proportional to the number of camera angle and illumination changes on the target face. As a reference, approximately 50 faces were extracted per scene. The dataset has then been doubled with real face images, also extracted from various internet sources and with the same resolutions. Finally, it has been manually reviewed to remove misalignment and wrong face detection. As much as possible, the same distribution of good resolution and poor resolution images were used in both classes to avoid bias in the classification task. Precise numbers of the image count in each classes as long as the separation into a set used for training and for model evaluation can be found in Table 2.

3.1.2 Face2Face dataset

Additionally to the Deepfake dataset, we have examined whether the proposed architecture could be used to detect other face forgeries. As a good candidate, the FaceForensics dataset [18] contains over a thousand forged videos and their original using the Face2Face approach. This dataset is already split into a training, validation and testing set. More than extending the use of the proposed architecture to another classification task, one advantage of the FaceForensics set is to provide losslessly compressed videos, which has enabled us to evaluate the robustness of our model with different compression levels. To be able to compare our results with those from the FaceForensics paper [18][19], we have chosen the same compression rate with H.264: lossless compression, 23 (light compression), 40 (strong compression). Only 300 videos were used for training out of more than a thousand. For the model evaluation, the 150 forged video and their original of the testing set were used[19]. Details about the number of extracted face images for each class can be found in Table 2.

3.2. Classification Setup We denote X the input set and Y the output set, the random variable pair (X, Y) taking values in X × Y, and f the prediction function of the chosen classifier that takes

Set	**forged** class	**real** class
Deepfake training	5111	7250
Deepfake testing	2889	4259
Face2Face training	4500	4500
Face2Face testing	3000	3000

Fig.11 Cardinality of each class in the studied datasets. Note that for both datasets, 10% of the training set was used during the optimization for model validation.

B. EfficientNet Compound Rate

EfficientNet is a method of constructing a convolutional neural network and a method of measuring equally all depth / width / adjustment measurements using a combined coefficient[20]. Unlike a standard practice that scales these features incorrectly, EfficientNet's measurement method measures network width, depth, and similar alignment with a set of unmodified scale parameters[21]. For example, if we want to use 2N counting resources, we can simply increase the network depth by αN, width βN, and image size by γN, where α, β, γ are static coefficients determined by a small grid search. in the original small model. EfficientNet uses a combination coefficient ϕ to accurately measure network width, depth, and optimization in a systematic way. The integrated measurement method is supported by the idea that when the input image is larger, the network needs additional layers to enlarge the reception field and additional channels to capture the best analyzed patterns in the larger image. The effective EfficientNet-B0 network is based on the remaining blocks of the MobileNetV2 bottle, in addition to the compression blocks and excitement. EfficientNets[20] also effectively transmits and achieves modern accuracy in CIFAR-100 (91.7%), Flowers (98.8%), and 3 other transfer data sets for order, with a few size parameters order.

Intuition behind the network

We have tried to understand how those networks solve the classification problem. This can be done by interpreting weights of the different convolutional kernel and neurons[18] as image descriptors. For instance, a sequence of a positive weight, a negative one, then a positive one again, can be interpreted as a discrete second order derivation. However, this is only relevant for the first layer and does not tell much in the case of faces.
We can also take the mean output of a layer for batches of real and forged images, observe the differences of activation and interpret the parts of the input images that play a key role in the classification. If we study the trained MesoInception-4[18] network on the deepfake dataset, as it can be seen in Figure 8, eyes are strongly activated for real images but not on deepfake images for which the background shows the highest peaks. We can surmise that it is again a question of blurriness: the eyes being the most detailed part of real images while it's the background in forged images because of the dimension[18] reduction underwent by the face.

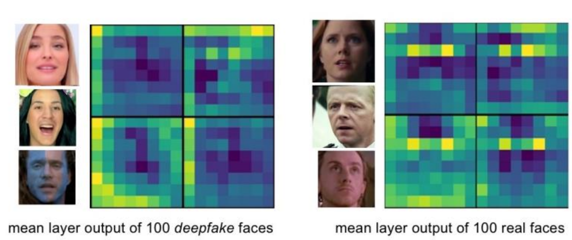


Fig.13 Mean outputs of the Deepfake dataset for the some filters of the last convolutional layer.

Experimental results:

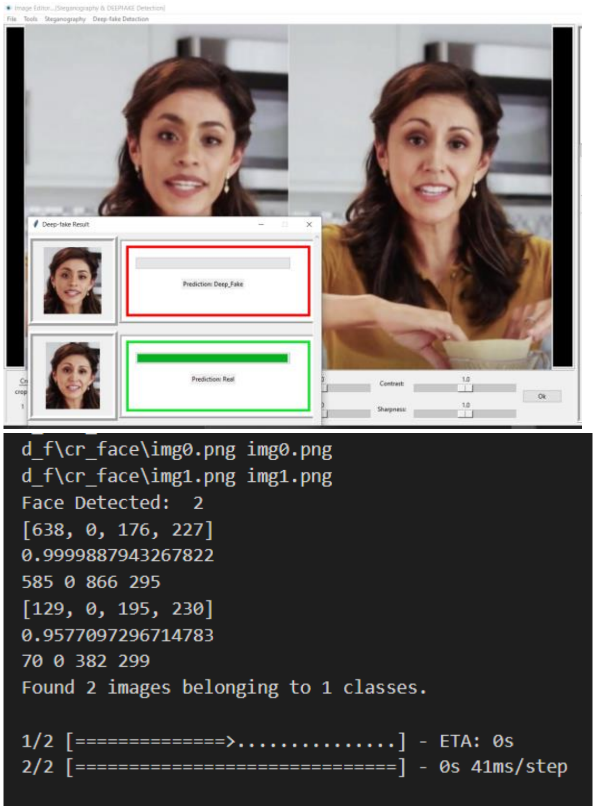


Fig .14 Result of our output

As shown in fig.14 the dataset is able to recognize the number of faces detected in the given given and by using MTCNN[18] the faces are are croped and so as our trained CNN[19] dataset compares the faces the result is shown whether the faces in the image is deep faked or not. As we have selected an image the result is shown for single image, as it checks how many faces does the given image have and thus the result of those images is shown. These is for the chosen image, We can even detect this for video also.

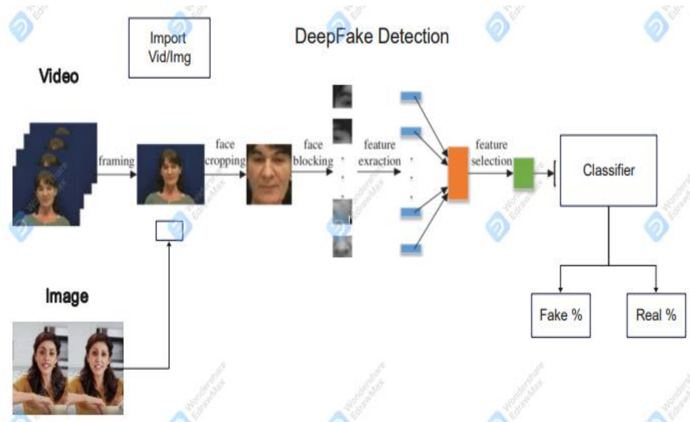
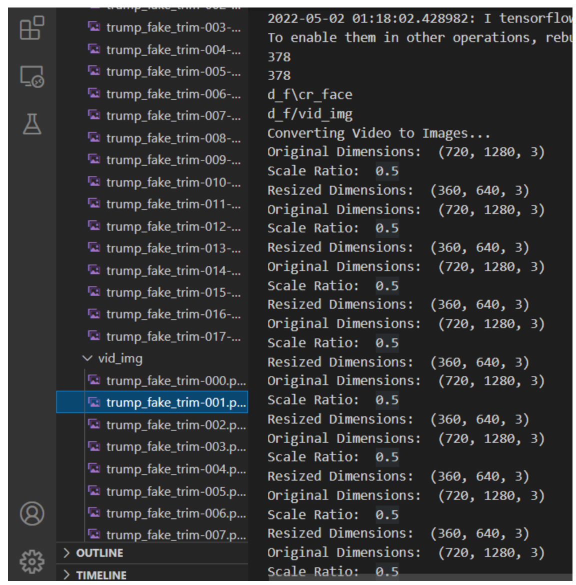
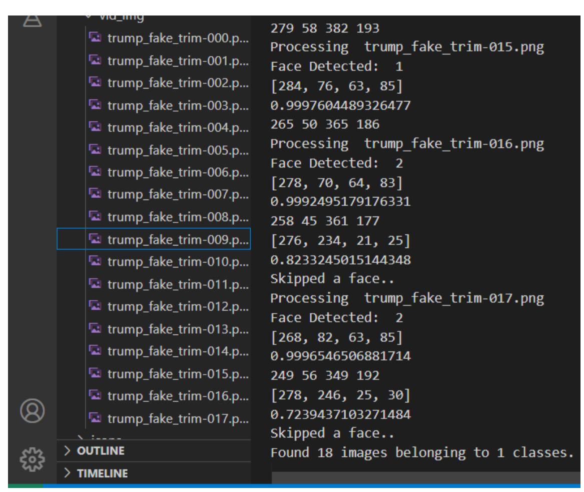


Fig . 15 Block Digram of Deepfake detection

For video the program first trims the video into several frames, as shown in fig. the given deepfaked video of Trump is trimed into 18 frames.





Then the trimmed images/frames trimmed from video under goes same procedure as for single image though our CNN model, first face is extracted using MTCNN[22] dataset and for each frame the the result is shown for eg. In given fig total 18 images are trimmed, so the total images extracted are from (0 – 17) . The model then process every frame and the result is shown of every frame whether is seemed to be deepfaked or not. Our GUI provides proper presentation of our result.

```
[276, 234, 21, 25]
0.8233245015144348
Skipped a face..
Processing  trump_fake_trim-017.png
Face Detected:  2
[268, 82, 63, 85]
0.9996546506881714
249 56 349 192
[278, 246, 25, 30]
0.7239437103271484
Skipped a face..
Found 18 images belonging to 1 classes.

 1/18 [>...........................] - ETA: 48s
 3/18 [====>.......................] - ETA: 0s
 5/18 [=======>....................] - ETA: 0s
 7/18 [==========>.................] - ETA: 0s
 9/18 [==============>.............] - ETA: 0s
10/18 [===============>............] - ETA: 0s
12/18 [===================>........] - ETA: 0s
13/18 [====================>.......] - ETA: 0s
15/18 [========================>.....] - ETA: 0s
17/18 [===========================>..] - ETA: 0s
18/18 [==============================] - 4s 44ms/step
d:\Mini  Project 2022\Github\Image-Detection-and-Manipulation-mas
  return imgt.resize((l, w), Image.ANTIALIAS)
```

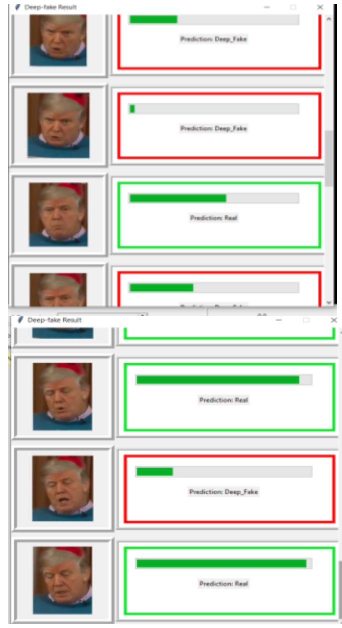


Fig 15 . Predcition of the video per frame

Block Diagram:

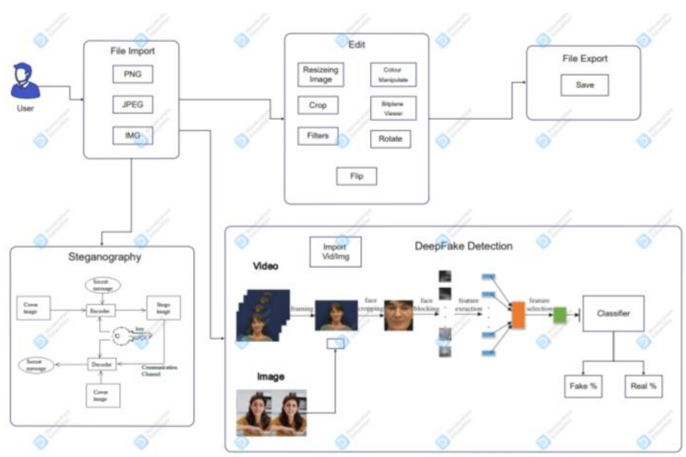


Fig 16. Complete Block Diagram.

Conclusion:

This paper demonstrates two structures for Steganography: Initially it is the striking rationality which is also known as Least Significant Bit(LSB), and the second one is the latest system with LSB+KEY. The results executions have been looked up for the estimations of PSNR with individual checks. It is seen that the [7] calculation of LSB+KEY gives better demands concerning the PSNR regards. This is one of the investigated results in this work and still the work is in its head-way to improve the computations for still better code unconventionality and time complexity nature. Also we presented a deep study for binary classification of deepfake videos. We analyzed different approaches to improve the video classification in high compression factor These days, the dangers of face tampering in video are widely recognized. We provide two possible network architectures to detect such forgeries efficiently and with a low computational cost. In addition, we give access to a dataset devoted to the Deepfake approach, a very popular yet under documented topic to our knowledge. Our experiments show that our method has an average detection rate of 98% for Deepfake videos and 95% for Face2Face videos under real conditions of diffusion on the internet. We have notably understood that the eyes and mouth play a paramount role in the detection of faces forged with Deep fake. We believe that more tools will emerge in the future toward an even better understanding of deep networks to create more effective and efficient ones. Also we learnt how to integrate our trained model with our GUI for a friendly user interface.

References:

[1] F. Hartung and M. Kutte "Information hiding- a survey, " Proceedings of the IEEE: Special Issue on Identification and Protection of Multimedia Content, Volume: 87 Issue: 7, pp. I062-I078, July. 1999.

[2] M. Hossain, S.A. Haque, F. Sharmin, "Variable Rate Steganography in Gray Scale Digital Images Using Neighborhood Pixel Iriformation", Proceedings of 2009 12th International Conference on Computer and Information Technology (ICCIT 2009) 21-23 December 2009, Dhaka, Bangladesh.

[3] Na-I Wu, "A Study on Data Hiding/or Gray-Level and Binary Images ", http://www.google.com/url?sa=t&rct=j&q=&esrc=s&sourc e=web&cd= 1 &ved=OCBwQFjAA&url=http%3A %2F%2F ethesys.lib.cyut.edu.tw%2FETD-db%2FETDsearch%2Fgetfile%3FURN%3Detd-0707104-44705%26filename%3Detd-0707104-144705.pdf&ei=yMavTr7LOoSBhQet3pHRAg &usg= AFQ jCNFztbb-TMOJ3fg_ Qvv8DsUDY8qwA ,Accessed on March 2009.

[4] GJ. Simmons, "The Prisoners' problem and the subliminal channel," in proc. CRYPTO'83, pp. 51-67, 1983.

[5] T. Morkel, J. H. P. Elloff, M.S. Olivier, ―An Overview of Image Steganography‖.

[6] Dr. Amarendra K, Venkata Naresh Mandhala, B.Chetan gupta, G.Geetha Sudheshna, V.Venkata AnushaImage-Steganography-Using-Lsb http://www.ijstr.org/final-print/dec2019/-

[7] Provos, N. &Honeyman, P., ―Hide and Seek: An introduction to steganography‖, IEEE Security and Privacy Journal, 2003.

[8] Akash Kumar MANAS & Arnav Bhavsar MANAS Detecting Deepfakes with Metric Learning march,2020.

[9] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. OrtegaGarcia, "Deepfakes and beyond: A survey of face manipulation and fake detection," 2020.

[10] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and ¨ M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," CoRR, vol. abs/1901.08971, 2019. [Online]. Available: http://arxiv.org/abs/1901.08971

[11]V. Schetinger, M. M. Oliveira, R. da Silva, and T. J. Carvalho. Humans are easily fooled by digital images. arXiv preprint arXiv:1509.05301, 2015.

[12] B. Balas and C. Tonsager. Face animacy is not all in the eyes: Evidence from contrast chimeras. Perception, 43(5):355– 367, 2014.

[13] S. Fan, R. Wang, T.-T. Ng, C. Y.-C. Tan, J. S. Herberg, and B. L. Koenig. Human perception of visual realism for photo and computer-generated face images. ACM Transactions on Applied Perception (TAP), 11(2):7, 2014

[14] S. Fan, R. Wang, T.-T. Ng, C. Y.-C. Tan, J. S. Herberg, and B. L. Koenig. Human perception of visual realism for photo and computer-generated face images. ACM Transactions on Applied Perception (TAP), 11(2):7, 2014

[15] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[17] Darius Afchar Ecole des Ponts Paristech ´ Marne-la-Vallee, France fr Vincent Nozick JFLI, CNRS, UMI 3527, Japan LIGM, UMR 8049, UPEM, France ,Junichi Yamagishi, Isao Echizen National Institute of Informatics Tokyo, Japan MesoNet: a Compact Facial Video Forgery Detection Network

[18] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, ¨ and M. Nießner. Faceforensics: A large-scale video dataset for forgery detection in human faces. arXiv preprint arXiv:1803.09179, 2018.