How to Cite:
Jayaram, D., Gopalachari, M. V., Rakesh, S., Sai, J. S., & Kumar, G. K. (2022). Fake face image detection using feature network. *International Journal of Health Sciences*, *6*(S5), 3027–3039. https://doi.org/10.53730/ijhs.v6nS5.9310

Fake face image detection using feature network

D Jayaram
Assistant Professor, IT Department, Chaitanya Bharathi Institute of Technology, Hyderabad, India

M Venu Gopalachari
Associate Professor, IT Department, Chaitanya Bharathi Institute of Technology, Hyderabad, India
Corresponding author's Email: mvenugopalachari_it@cbit.ac.in

S. Rakesh
Assistant Professor, IT Department, Chaitanya Bharathi Institute of Technology, Hyderabad, India

J Shiva Sai
Assistant Professor, CSE Department, Chaitanya Bharathi Institute of Technology, Hyderabad, India

G Kiran Kumar
Assistant Professor, CSE Department, Chaitanya Bharathi Institute of Technology, Hyderabad, India

***Abstract*---**In the recent times, the image data in social networks such as Instagram, Whatsapp, Facebook, Snapchat, twitter etc has an exponential growth in terms of volume, variety due to the velocity of the data stream. On the other hand, the advancements in the image and video processing led to increase in the fake images relatively in huge volumes. Due to the involvement in spreading fake news and leading mob incitements fake images became major concern to handle that demands an efficient a fake image detector is of at most concern to entire social networking organizations. In this paper, a deep learning framework is proposed that differentiates fabricated parts of the image from the real image using supervised learning strategies. Also a modified neural network structure called the Fake Feature Network is proposed in this work which consists of advanced convolution networks. In order to make model effective, the proposed methodology has a two major steps in learning which combines a modified neural structure that uses classifier and pairwise learning for the fake image detection. The performance of

the model is enhanced the contrastive loss to learn the common fake features using pairwise learning, which is achieved by incorporating Siamese neural network.

***Keywords*---**Fake face image detection, Feature Network, Social Network Data, Deep learning, Pairwise learning.

1. Introduction

In this technological era, social media networks playing good role in human's life in all aspects. Over the years we have seen social media take different forms than it was created for. One of the problematic forms of social media is spreading fake information and misinformation. People started using social media as a tool to spread fake information to exploit others or damage their reputation. One of the main ways of doing this was creating fake images or videos. The major contributors of creating fake images are Generative Adversarial Networks aka GANs.

GANs are deep learning based generative models that have been widely used to create partial or whole realistic images or videos. These neural networks have been progressing every year creating more sophisticated networks like BigGAN, CycleGAN and PGGAN, which create highly realistic images. They also have the capability to create synthesized speech videos, which was used in the past to sabotage several politician's credibility.

Current forensic techniques require an expert to analyze the integrity of an image. Since the fake images created are highly sophisticated and almost impossible to detect a system is needed which is on par to the latest GANs models to detect them. A model using customized advanced neural networks to determine fake images has been implemented. A modified neural network structure called Fake Feature Network (FFN) has been proposed, which consists of Convolution Neural Networks, DenseNETs and Siamese Neural Networks. The model created will be trained using pairwise learning strategy to obtain an efficient model.

2. Related Work

There exists different kind of tools and techniques to handle digital image forensics in the literature [10,11]. In order to justify the image authenticity majority of the approaches focus on the format, Meta data of the image data. Though the advancements in this category of tools and techniques targeting fake images, detecting the fake images remains as a challenge due to the availability of the same advancements of the image processing techniques for the attacker. Few reputed social networking organizations such as facebook and Microsoft are sponsoring the fake detection challenge program in recent time to tackle this challenge [12].

One of the traditional approaches for fake image detection is frequency domain analysis that focuses on the compression data of the image that reveals different values for manipulated images. The fake images of faces usually tend to contain

anomalies when compared to real ones in frequency domain and the experiments in the literature with deep fake image detector had exhibit good results on the image spectrum [13]. Due to the existence of smooth and sophisticated edges in the image the frequency domain based fake image detectors could not perform upto the mark. To address these limitations in this category, few techniques such as JPEG Ghost proposed in the literature [14], in which the JPEG quality of different regions on the same image identified and verified. If the JPEG quality of all regions is not same, then the image assumed to be fake as it was recreated by incorporating another image region. But if the forged image maintains the same JPEG quality on all regions of the image, JPEG Ghost like techniques will not perform well.

On the other hand few researchers focused on the texture features of the image to differentiate fake from real images. However, designing a fake image detector based on the texture information seems critical as the sole global texture information could not differentiate fake images from real images [10]. As as advancement in this approach gram matrix is introduced to capture texture information combined with Resnet [15]. If the images manipulated are of format JPGE, which is a lossy format, one can easily recognize the regions manipulated using Error Level Analysis. JPGE images maintains stable areas throughout the image which have minimum error. Any modification done to a picture will result in altering of stable areas. ELA tries identifies the areas manipulated since they are no longer at their minimum error level. By analysing the patter of the images after ELA once can determine which parts of the image are possibly faked. One can integrate machine learning to detect the anomalies in the error level analysed images too. If the different regions of the image are manipulated then this approach suits better whereas error level analysis is not efficient for GAN images.

Deep learning is also another latest approach in handling fake face detection techniques. Convolution Neural networks is the popular deep learning technique spread over various domain also can be used for face forgery methods such as FaceForensics[3], in which convolution and max pooling are the two major steps driving the supervised learning process. In this category of methods, self-attention generative adversarial networks are other technique which generates fake face data set which is refined [4].

In [6] and [7] authors proposed LSGAN (Least Square Generative Adversarial Networks) which improves the training model performance for fake image detection by supplying additional features such as noise features which are generated by steganalysis, as input to the CNN model. However, if the images have smooth edges then due to the challenges in extracting the noise feature, this kind of approaches cannot perform well for fake image detection. Few approaches in the literature such as MMC image forensic tool uses JPEG quality and metadata features [9]. Fotoforensics is the other tool that uses SNGAN (Spectral Normalization for Generative Adversial networks for fake image detection [8]. But the challenge lies with the images provide by GAN technique as well if the meta data itself is modified or hidden by the attacker, these kind of tools fail to identify fake images.

In [5], a self-consistency based learning method on social network data is proposed in which it justifies whether a single imaging pipeline could generate the content or not that reuires the EXIF meta data. However this approach is also sensitive to the attacks on meta data of the fake images. The model proposed by "Fake image detection using machine learning" [2] To detect GANs generated images uses shallow networks instead of deep neural networks. They also use an adversary model to delete the entire metadata. The main limitation of their model is that they manually created fake face images to create the model, which is quite time consuming.

3. Proposed FFN based Model

The data is first preprocessed where the images are reshaped to 64 x 64 pixels resolution. Pairs of images from these reshaped images are made which are necessary for pairwise learning in Siamese networks. The Siamese network is built using the Fake Feature Network as its sister networks. The Siamese network is trained using contrastive loss function. Finally testing and validation is done using various validation methods. The working has been represented as shown in Fig. 1.

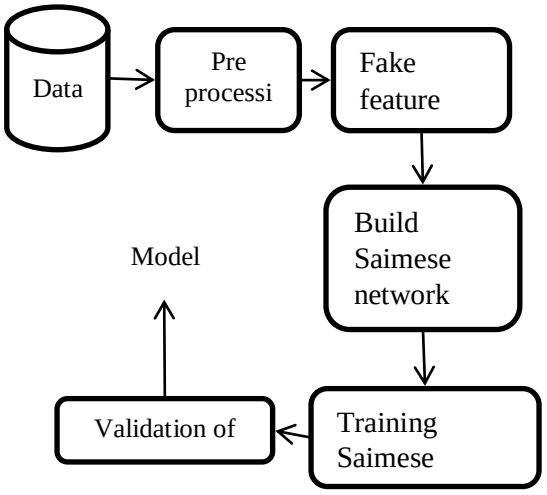

Figure 1 The architecture of the proposed model

A. Processing data

The images from the dataset were loaded to the model. Using an inbuilt function which images are loaded the from the directory. For this the directory was organized in such a way that it consisted of two folders which contain images of each class respectively i.e., a folder with all real images and a folder with all fake images. The images loaded were reshaped to 64x64 pixels and the number of images loaded once per call, i.e., batch size was 16.

B. Building Fake Feature Network

The architecture of the sister network is shown in Fig 2. The sister network initially consists of a convolution layer with 7 x 7 kernel size and 48 channels with stride value being 4. The output of this layer is sent to dense blocks DenseNets with different number of blocks in each unit are used to construct the Fake Feature Network Model (FFN). A total of two dense units containing 2 and 3 blocks, respectively are created. Each unit has 24 and 30 channels, respectively.

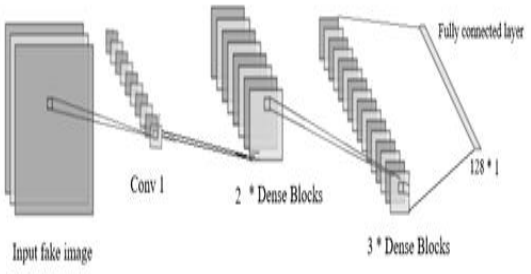

Figure 2 Design of Fake Feature Network

A translation layer is added in between every dense block, where the input tensor is halved. The dense block consists of batch normalization, ReLu activation and convolutions. Convolution with k * 4 filters and 1 x 1 kernel size are used, where k is the growth rate. Consecutively another batch normalization, ReLu and convolution is done. This time convolutions are done with k number of filters and 3 x 3 kernel size. Two sister networks are trained using discriminative feature learning. Following these is a fully connected layer of 128 neurons i.e, the output of the first dense unit is concatenated with the final output of the last dense unit and is sent to the fully connected layer.

C. Building Siamese neural network using FFN

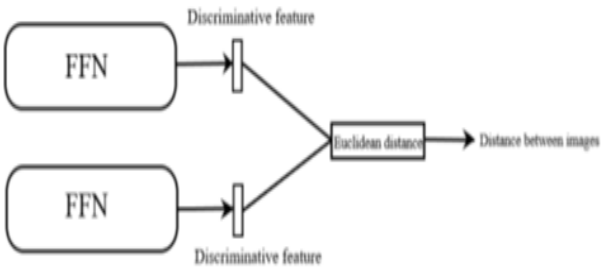

Figure 3 Siamese network architecture

Fig. 3 depicts the architecture of Siamese network. Sister networks have same architecture and parameters. Both networks work and train simultaneously to enable pairwise learning. A pair of images is given to the model where each sister network takes one image from the pair as input. The sister networks then give

extract discriminative features as output, which is passed to a function that calculates Euclidean distance between the features. If the distance between two images is less than 0.5, they belong to the same class else they belong to different classes.

D. Discriminative Feature Learning

To enhance the performance of the system, contrastive loss was introduced to learn the common fake features. Contrastive loss is extensively used in Siamese neural networks to calculates the distance between two images in vector space. The loss is low if the images are closer and high when images are farther apart. To achieve this, we incorporate contrastive loss in energy function of traditional loss function. The energy function defined as follows:

$$Ew(x1,x2) = ||fFFN(x1) - fFFN(x2)||22$$

Where (x1, x2) is the face image pair.
After incorporating contrastive loss into the function, it will be defined as:

$$L(W,(P,x1,x2)) = 0.5 \text{ x } (yijE2w) + (1 - yij) \text{ x } \max(0,(m-Ew)22)$$

Where y indicated pairwise label, that y=0 indicates an imposter pair and y=1 indicates a genuine pair. m denoted the predefined threshold. When input is genuine, the cost function minimizes the energy by making (1 - yij) x max(0, (m - Ew))22) equate to zero. When the input pair is an imposter pair the contrastive loss will maximize energy by minimizing function max (0, (m - Ew)). In this way by iteratively training the network using contrastive loss, common fake features of collected GANs can be learned.

4. Results and Validating FFN

N-way one shot learning is used to validate the Siamese neural network. The algorithm works in such a way that same character is compared to n different characters out of which only one of them matches the original character. The model must give minimum distance to the one that matches the parameter when compared to the rest. If so, it is treated as a correct prediction. This procedure is repeated k times, and the percentage of correct predictions is calculated as follows.

$$percent\_correct = (100 \text{ x } n\_correct) / k$$

where k represents the total number of trials and n_correct represents the number of correct predictions out of k trials.

The proposed model has been tested using N-way one shot learning technique. Where n number of pairs will be sent to the model out of which only one pair of images belong to the same class. The model is tested k number of times to check how accurately it predicts that one pair. To test this model 2,4,16 and 32 pairs of images have been tested 100 times for both real and fake images. The results are shown in Table 1.

Table 1 N-way one shot results

Class	Number of pairs	Repetitions	Accuracy
Real	2	100	100
Real	8	100	100
Real	4	100	100
Real	16	100	100
Real	32	100	99
Fake	2	100	100
Fake	4	100	100
Fake	8	100	100
Fake	16	100	98
Fake	32	100	98

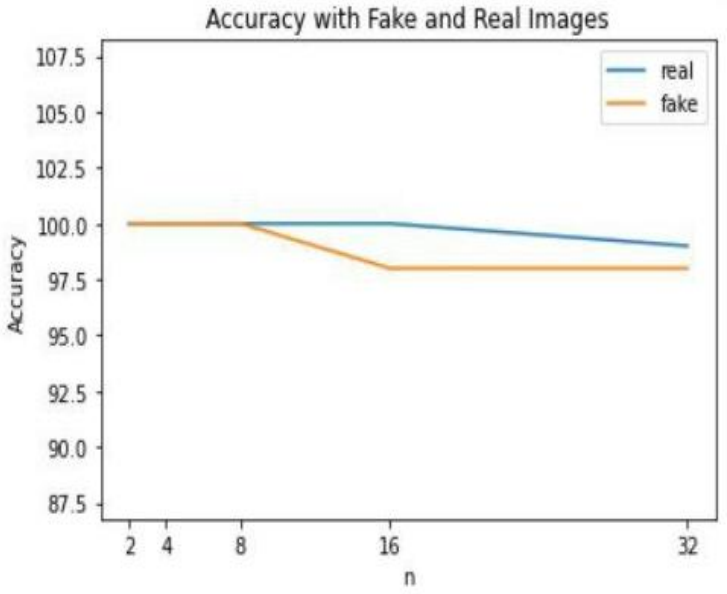

Figure 4 Accuracy curve

The model has also been tested by changing the batch size of input given to the model. Time taken in training has also been added as a constraint to test the performance of the model. The results are shown in Table 2.

Table 2 Test results of proposed model

Batch size	Time taken per epoch	Loss
8	6 hrs	0.0017
16	3.5 hrs	0.0017
32	2 hrs	0.0018
64	1.5 hrs	0.0020

To get the model with low training time per epoch and loss, models with 8,16,32 and 64 batch sizes were test and are show in Table 2. Based on our testing

process, it was found that modified neural model with 8 batch sizes performed better when compared to the rest.

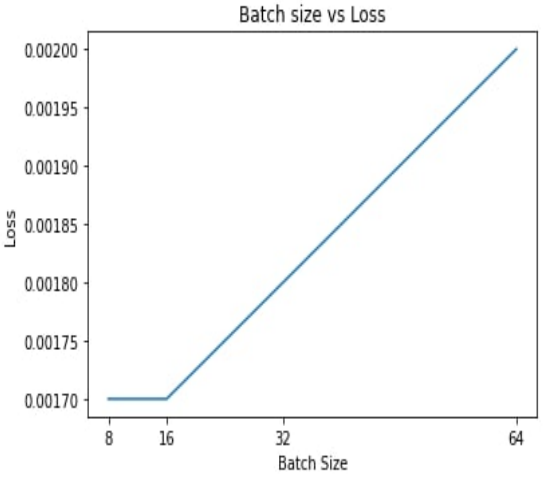

Figure 5 Batch size vs loss curve

On evaluating the model built over the validation data, the accuracy and loss obtained were 99.5% and 0.0017, respectively.

Figure 6 Output when training

Random Model

Random model is a common technique which is used while testing neural network incorporated with Siamese neural networks. Random model working is similar to that of N-way one shot learning. It classifies the image given as input by comparing it to 16 random images. The set of images it uses to compare the test image is called Support Set. The model then generates n random numbers between 0 and 1. n random numbers denote the similarity score between the test image and one of the images in the support set. While testing, if the model gives a maximum similarity score to the one it is like, then it is considered as correct prediction.

The model created was also tested with a Random model. A random model was created and tested using 2,00,000 images using different n values. 2, 4 8,16 and 32 n values were used to test the model, where each n value prediction was repeated 100 times.

Table 3 Test results for random model

N Value	Number of trials	Accuracy
2	100	44
4	100	25
8	100	16
16	100	7
32	100	5

Figure 7 Random model vs FFN

The model was also tested with a random model. Table 3 represents the results of the random model. The results were compared to the model proposed and the results are show by Fig 7. Random model`s performance deteriorated as the number of numbers it generated increased. Whereas the proposed model`s performance was almost linear when batch size increased. After comparing the results of both model FFN model proved to be more efficient.

Executable model

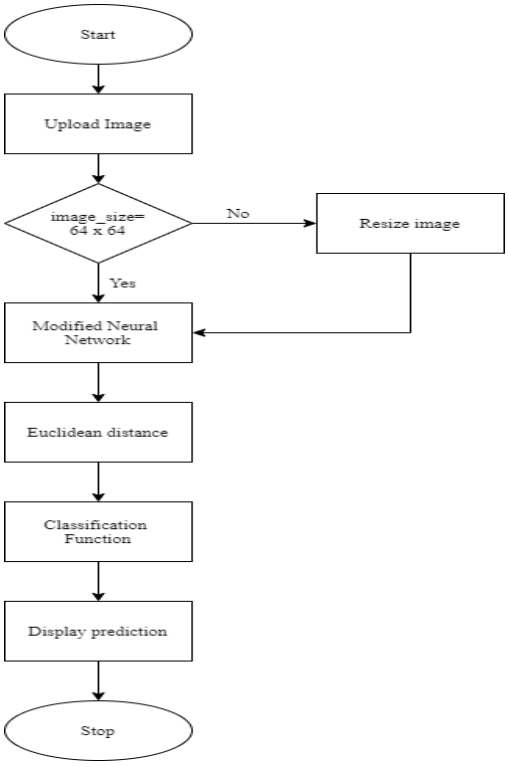

Figure 8 Flow chart of execution

Upload Image

Once the user executes the application, a Graphical User Interface pops up on the screen. The GUI prompts the user to upload an image which he wants to classify. The dialogs and buttons on the GUI were constructed using Tkinter library in python.

Figure 9 Screenshot of user Interface

Upon clicking the upload image button, the user is directed to another window which allows him to choose his picture. The selected image`s path is returned and passed to the modified neural network.

Figure 10 Screenshot of uploading image

Resize Image

Dimension of the image uploaded by the user are first checked before sending it to the modified neural network. The input image is passes as it is if its is 64 x 64, else the image is resized using cv2.resize function which is a part of python OpenCV library. The resized image is then passed to the neural network.

Modified Neural Network

The images given by the user are sent to the FFN. Along with the input image a set of n fake and real images are randomly picked and sent to the FFN. FFN extracts discriminative features from the input image and the set of n images given to is. The extracted features are stored in a feature vector. Feature vector is then sent to the lambda layer.

Euclidean distance

The lambda layer calculates Euclidean distance between the input image and the set of n images it randomly chose, using the feature vectors given by the user. The Euclidean distance represents the distance between two images in a vector space. If the two images belong to the same class their Euclidean distance is less than 0.5 and more than 0.5 when they belong to different classes.

Classification Function

The classification used works with a simple logic and uses simple python language. When the Euclidean distance is passed to the function it tries to check how close it is to the random picture it is being compared to. If the distance between them is less than 0.5 the random image label is checked, and the input image is classified in the same class as the random image. If the distance between

them is higher than 0.5 the input image is classified in a different class than that of the random image`s.

Display Prediction

Once the image is classified the class name is shown on the GUI through create_text function present in Tkinter library of python.

Figure 11 Screenshot of prediction

5. Conclusion

A novel modified neural network for fake face detection is proposed in this paper. Since the model has only been trained using single face images it gave satisfactory results with images with multiple imaged were given as input. The ability of this system can be extended to detect fake videos and images with more than one face. One can detect fake news or videos and assures the credibility of the data. Incorporating face detection will help in case of images with multiple faces. The model can used face detection to identify the faces in the image and classify them individually.

References

[1] Chih-Chung Hsu, Yi-Xiu Zuhang and Chia-Yen Lee. “Deep Fake Image Detection Based on Pairwise Learning”, Published in MDPI, Jan 3 2020

[2] Muhammed Asfal Villain, Johns Paul, Kuncheria Kuruvilla and Eldo P Elias. “Fake Image Detection using Machine Learning” In proceedings of IEE, Mar-Arpil 2017.

[3] Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J.” Is object localization for free?-weakly-supervised learning with convolutional neural networks.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.

[4] Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. “Self-Attention generative adversarial networks”. In Proceedings of the 36th International Conference on Machine Learning; Chaudhuri, K., Salakhutdinov, R., Eds.; PMLR: Long Beach, CA, USA, 2019

[5] Marra, F.; Gragnaniello, D.; Cozzolino, D.; Verdoliva, L. “Detection of GAN-Generated Fake Images over Social Networks.” In Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval, Miami, FL, USA, 10–12 April 2018.

[6] Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. “Improved training of wasserstein gans.” In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017.

[7] Mao, X.; Li, Q.; Xie, H.; Lau, R.Y.; Wang, Z.; Smolley, S.P. “Least squares generative adversarial networks”. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.

[8] Miyato, T.; Kataoka, T.; Koyama, M.; Yoshida, Y.” Spectral normalization for generative adversarial networks.” In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.

[9] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.;Bernstein, M.; et al. “ImageNet large scale visual recognition challenge.” Int. J. Comput. Vis. (IJCV) 2015, 115, 211–252.

[10] Zhengzhe Liu, Xiaojuan Qi, Jiaya Jia and P. Torr, "Global texture enhancement for fake face detection in the wild", 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8057-8066, 2020.

[11] Xiaodan Li, Yining Lang, Yuefeng Chen, Xiaofeng Mao, Yuan He, Shuhui Wang, et al., "Sharp multiple instance learning for deepfake video detection", Proceedings of the 28th ACM International Conference on Multimedia, Oct 2020.

[12] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram and Cristian Canton Ferrer, "The deepfake detection challenge (DFDC) preview dataset", arXiv preprint arXiv:1910.08854 [cs.CV], 2019.

[13] [Luca Guarnera, Oliver Giudice, Cristina Nastasi and Sebastiano Battiato, "Preliminary forensics analysis of deepfake images", arXiv preprint arXiv:2004.12626 [cs.CV], 2020.

[14] Rinartha, K., & Suryasa, W. (2017). Comparative study for better result on query suggestion of article searching with MySQL pattern matching and Jaccard similarity. In *2017 5th International Conference on Cyber and IT Service Management (CITSM)* (pp. 1-4). IEEE.

[15] S. Azarian-Pour, M. Babaie-Zadeh and A. R. Sadri, "An automatic JPEG ghost detection approach for digital image forensics," 2016 24th Iranian Conference on Electrical Engineering (ICEE), 2016, pp. 1645-1649.

[16] Humeau-Heurtier, "Texture feature extraction methods: A survey", IEEE Access, vol. 7, pp. 8975-9000, 2019.