Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.777719

AN ANALYSIS ON IMPROVING DEEPFAKE DETECTION: AN IMAGE BASED APPROACH

Anushree Deshmukh[1] and Sunil Wankhade[2]

[1]Research Scholar, Computer Engineering Department, Rajiv Gandhi Institute of Technology, RGIT, Andheri, Mumbai-53, anushree.deshmukh@mctrgit.ac.in

[2]Professor, Department of Information Technology, RGIT, Andheri, Mumbai-53, sunil.wankhade@mctrgit.ac.in

Abstract - Technology has advanced so much that anyone can edit images with the software available and converts the real picture into fake picture. Selfies have become an integral part of photography in recent years, and they are even considered a powerful and trustworthy medium of communication. To identify fake faces in various systems, many spoof detection techniques have been created, but face forgery continues as a challenge in social media platform. This paper specifies a robust algorithm that can detect fake faces in shared photographs on social media. A Haar cascade classifier is used for feature selection and CNN for classification whether the input image is real or fake. This paper will analyze the existing system with the current accuracy and how our method reaches the highest accuracy for deepfake detection.

Key Words: Haar Cascade Classifier, Convolutional Neural Networks, Face Spoofing

INTRODUCTION

Photographs are used everywhere these days. It is very easy and common to manipulate the images using free applications, it won't leave any noticeable traces of those changes. This makes identifying the authenticity of an image extremely challenging [3], just by looking at an image, it's impossible to tell which is original and which is fake. Most of what you see on social media isn't real. Although social media is intended to socialize, share, and spread knowledge, there are people who misuse this platform to spread false information. Such manipulation of images has made identifying the true images very difficult [4]. The technological age has seen an enormous number of individuals becoming victims of photo forgery. The courts of justice are sometimes confused by criminals who use software to use pictures as evidence and take advantage of the system. To eliminate the current problem, all photographs shared via social media should be labeled real or fake [7]. As a result, forgery detection is a critical field for determining the validity of a photograph, as photographs are used as evidence in a variety of situations. A wide range of spoofing methods are presently being developed and used in fingerprint systems [16]. This study will be beneficial in monitoring social media campaigns and detecting fraud on social networking sites, particularly in the field of image-related social networking. With the substantial accessibility of pictures and videos in social contents Deepfake become popular [1]. This is mostly important now a days because the software for creating deepfakes are becoming more available, and anyone can share this fake content on social media without any restrictions. Deep learning algorithms has gained attention in many areas [8]. In recent times, various deep learning-based algorithms have been suggested to address the concern and successfully detect forged images and videos. The Convolutional Neural Network (CNN) is taught to distinguish between real and fake faces. It is made up of numerous hidden layers, including a convolutional layer, an activation function, a pooling layer, and fully linked layers between the source and final output layers. Hidden layers consist of neuron which studies the attributes of the input pictures and then forecast the classes, which are real and spoof. The output layer contains a sigmoid activation function. If the value from the output layer is less than 0.5 then classify the image as fake as it is the first class of images and if the value is greater than 0.5, classify the images as real.

In this paper, we have discussed complete explanation of the existing deepfake methods based on architecture, performance, and current challenges. Despite the fact that deep learning has been successful in detecting deepfakes, the quality of fake videos is increasing [2].

LITERATURE SURVEY

Earlier face spoofing detection primarily relied on motion, texture, frequency, and quality parameters to detect real and fake faces. Very little work has been finalized around detecting forgery audio, images, and videos [6]. However, several studies and tasks have been devised to spot what is often done around the enormous proliferation of counterfeit images online[17]. Adobe understands how Photoshop is misused and has tried to offer a cure.

Table 1. Analysis of Deepfake detecting methods.

Methods	Classifiers/Techniques	Features	Dealing with	Dataset Used	Findings
Eye blinking [5]	LRCN	LRCN is used to check the temporal dependencyin blinking pattern.	Videos	Dataset with 49 videos	A method can be developed for closed eyes. Only uses the lack of blinking as a cue fordetection.
Spatio temporal Features [10]	RCN	Temporal inconsistenciesare extracted using RCN that integrates convolutional network Dense Network	Videos	Face Forensics++	To explore how to increase the robustness ofour system against manipulated videos usingunseen techniques during training.
Using face wrapping artifacts[12]	VGG16 ResNet50	Artifacts are discoveredusing CNN model	Videos	UADFV, containing49 real and 49 fake videos.	For more efficient detection, a dedicated network structure for the detection of DeepFake videos is required.
MesoNet [13]	CNN	Meso-4 and MesoInception-4	Videos	Face Forensics and Face2Face	More tools will emerge inthe future toward an even better understanding of deep networks to create more effective and efficient ones.

Capsule forensics[14]	Capsule Network	Features are extracted by VGG-19 network is feededto the capsule network for classification. Two output capsules, one for fake and another for real images, is created by three capsule network andan algorithm is used to direct the output through a number of iterations.	Videos/ Images	Two data sets: Face Forensics, fully computer-generated image set	This method imitates representation and thelack of landmark adaptation. Personality mismatch canbe identified using landmarks from a different person.

Literature survey as shown in table 1, it has been observed that, in order to identify fake videos and images, deep learning methods need to improve. [16]. There is currently no clear way to know how many layers are needed for deep learning and which algorithm is correct. Additional region of research focuses on incorporating identification methods into social media platforms so they can be more effective in combating the effects of deepfakes and reduce its negative consequences. Based on table 1, and the literature survey it has been identified that the existing system has certain limitations. The dataset used in many papers are very small in size as of it contains only 49 videos. Though the study reaches the accuracy, but size of database will affect the performance of the system. There are many such finding after the survey, it has been observed that the model works good for only open eyes. It will not respond for the images or videos where the eyes of person are closed, it only works on a single clue of blinking of eyes. If the frequency of eye blinking varies; the model will decide whether the video is input video is real or fake. It Specifies the use of different artifacts which are trained using VGG16 and ResNet but here again the size of dataset is very small, it includes only 49 videos out of which most of the videos are compressed. Moreover, model works on capsule forensics where the limitation is these methods are the mimic representation and lack of landmark identification. Different landmarks from different people will result in personality mismatch.

PROPOSED SYSTEM

The Accuracy gained by the existing model is around 0.83. The main aim of the proposed model is to increase the accuracy of the model and present good results. Spoofing occurs when someone attempts to impersonate a registered user to gain illegal access and benefit from the protected system. Intruders present forged image for the illegal access into the system. To identify whether the image is fake or real we are proposing a novel approach in two stages:

- Face detection in the image.
- Face Validation of real or fake faces in the image.

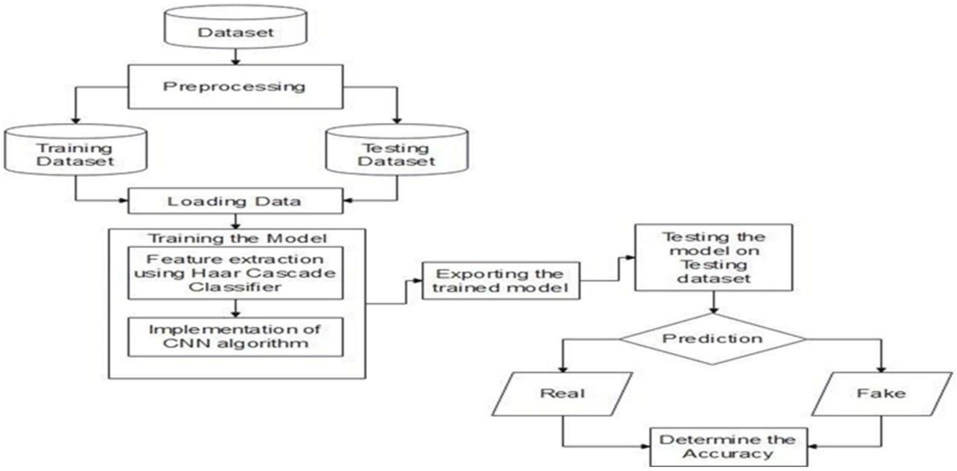

Dataset
Preprocessing
Training Dataset
Testing Dataset
Loading Data
Training the Model
Feature extraction using Haar Cascade Classifier
Implementation of CNN algorithm
Exporting the trained model
Testing the model on Testing dataset
Prediction
Real
Fake
Determine the Accuracy

Fig1. Process Flow Diagram

3.1 IMPLEMENTATION

Step 1: Gathering of data: There are two types of images in the data set: fake faces and real faces. There are approximately 10,000 images each for both sets of images.

Step 2: Pre-processing of data:

The Haar Cascade Classifier is used to recognize faces. This method is effectively used for detecting objects using Haar cascade feature-based classifiers. The approach is a machine-learning-based one that is trained from a substantial number of real and fake images. These functions are then used to detect objects in other images. By using this classifier, we retain only the faces that satisfy certain conditions as shown in fig 2. This way, the images have been cleaned. The processed data will be used to build our model. Faces were recognized in each frame and categorized as real or phony, resulting in the creation of two folders. As a result, our data collection is made up of both fake and real data as shown in fig 1.

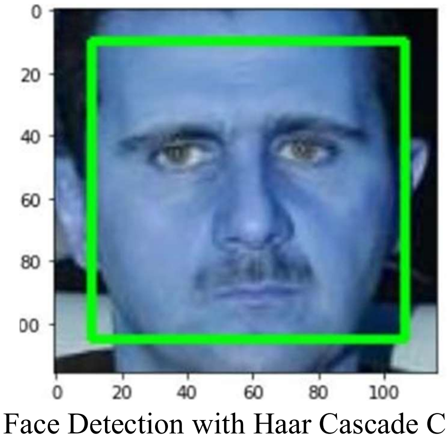


Fig 2. Face Detection with Haar Cascade Classifier

Step 3: Implementation of CNN model

The Convolutional Neural Network (CNN) is taught to distinguish between real and fake faces. It is made up of numerous hidden layers, including a convolutional layer, an activation function, a pooling layer, and fully linked layers between the source and final output layers as shown in fig.4 in the form of summary. Convolution Layer is the building block of CNN. This is the layer where most of the computations occur. This layer has 3 inputs. The first is input data, the second is a filter and the third is a feature map. The convolution process occurs on input data as shown in equation (i) that is the image using filters.

$$\text{i/p} = a^{[l-1]} \text{ withsize } \left(n_H^{[l-1]}, n_W^{[l-1]}, n_C^{[l-1]}\right), a^{[o]} \qquad \text{............ (i)}$$

Filters are nothing but a 3×3 matrix that is iterated over the input data which is also a matrix. As the filter is fed over the area of the input matrix, there is an activation function that calculates the result of the filter and the area of the image. The output is then stored in the feature matrix as shown in equation (ii).

$$\begin{gathered}\text{o/p} = a^{[l]} \text{ with size } \left(n_H^{[l]}, n_W^{[l]}, n_C^{[l]}\right) \text{ And we have} \\ \forall n \in \left[1,2, \ldots . n_C^{[l]}\right] \\ \text{conv}\left(\mathrm{a}^{[l-1]}, K^{(n)}\right)_{x,y} = \\ \psi^l\left(\sum_{n^n=1}^{n^{l-1}} \sum_{j=1}^{n^{l-1}} \sum_{k=1}^{n^{l-1}} K_{i,j,k}^n a_{x+i,y+j-1,k}^{[l-1]} + b_n^l\right) \qquad \text{..............(ii)} \\ \dim\left(\text{conv}\left(\mathrm{a}^{[l-1]}, K^{(n)}\right)\right) = \left(n_H^l, n_W^l\right)\end{gathered}$$

Pooling Layer is used for down sampling that reduces the number of parameters in our input image. It works in a similar way to the convolution layer. The only difference is that the filters in convolution layers contain some weights on which computation is performed by activation function whereas in the pooling layer we either take the max value in the input matrix or we take the average value in the matrix. There are two types of pooling: Max Pooling– In this pooling, we take the maximum of all the values in the patch of the matrix. Average Pooling– In this pooling, we take the average of all the values in the patch of the matrix. Fully Connected Layer, finally all the calculations we have done in the previous steps will pay off in this step as shown in equation (iii).

$$\begin{gathered}a_{x,y,z}^l = \text{pool}\left(a^{l-1}\right)x, y, z = \phi^l\left(a_{x+i-1,j-1,z}^{l-1}\right)i, j\varepsilon\left[1,2,3 \ldots f^l\right| \\ \dim\left(a^l\right) = n^l H, n^l W, n_C^l \qquad \text{..................(iii)}\end{gathered}$$

This layer finally classifies the image. In this layer, our input matrix is flattened, and it is passed through hidden layers of the neural networks. The previous Layers use ReLu activation functions for all computation while this layer uses the Sigmoid function for our binary classification problem.

So, this is how our model is working. The neurons in the hidden layer study the attributes of the input pictures and then forecast the classes, which are real and spoof. If the value from the output layer is less than 0.5 then we classify the image as fake as it is the first class of images we have and if the value is greater than 0.5, we classify the images as real.

In our case we have used three convolutional layers and ReLU activation function. There are three max-pooling layer of size 2x2, after the convolutional layer and dropout layers are also added to prevent overfitting. To classify, one completely connected dense layer is utilized, and the Sigmoid classifier is used.The data set is split into two parts: 80 percent for training and the remaining percent for testing. Real or Fake face detection is carried out using CNN architecture in fig.3.

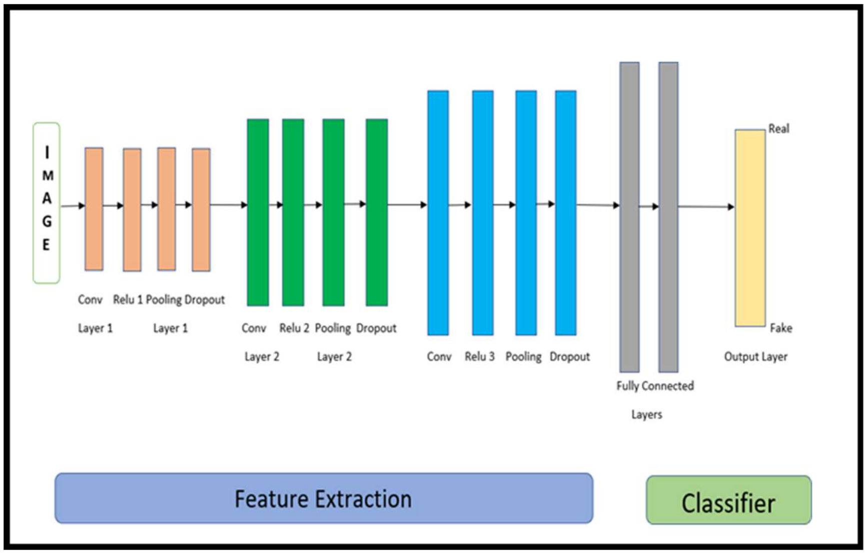
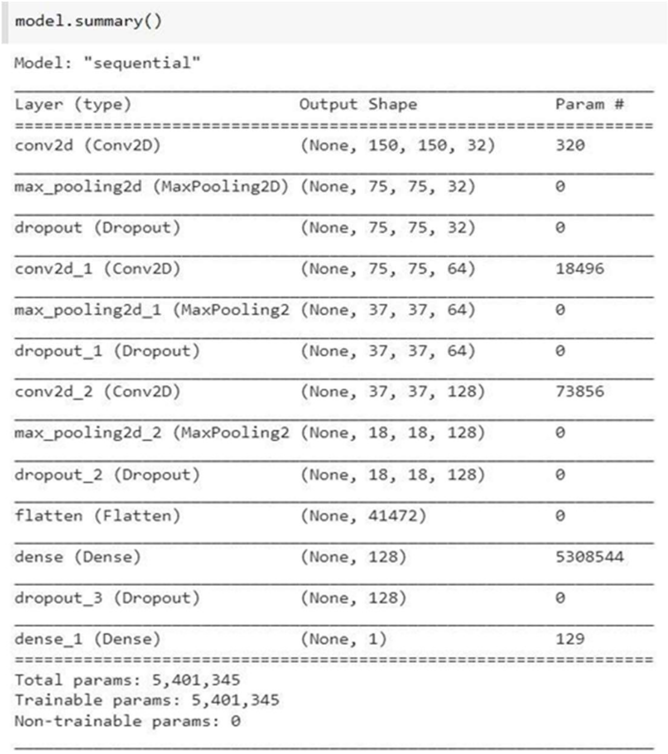


Fig 3. CNN Architecture



```
model.summary()

Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 150, 150, 32)      320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 75, 75, 32)        0
_________________________________________________________________
dropout (Dropout)            (None, 75, 75, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 75, 75, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 37, 37, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 37, 37, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 37, 37, 128)       73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 18, 18, 128)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 18, 18, 128)       0
_________________________________________________________________
flatten (Flatten)            (None, 41472)             0
_________________________________________________________________
dense (Dense)                (None, 128)               5308544
_________________________________________________________________
dropout_3 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 129
=================================================================
Total params: 5,401,345
Trainable params: 5,401,345
Non-trainable params: 0
_________________________________________________________________
```

Fig 4.CNN Model Summary

Step 4: Training – Phase and Testing – Phase

There are two phases namely, Training – Phase and Testing – Phase.

Training model: The training phase includes a dataset that includes both legitimate and malicious pictures. The cropped photos of faces are 150X150 in size. To train the dataset, the Convolutional Neural Network Model, as described above, is employed. The network is trained following image augmentation, which introduces some variety into the dataset.

Testing model: While testing our model we have drawn two graphs i.e., Accuracy vs Epochs and Loss vs Epochs. First graph shows how accuracy varies with the number of epochs as shown in fig.4 and second how loss varies with change in number of epochs as shown in fig.5.

RESULTS

Table.2 shows the result of training, testing accuracy and loss respectively. As the accuracy achieved in the existing system based on the model trained and the dataset used reaches to average. This proposed model gives the accuracy of 0.97 with a minimum loss of 0.67. Thus, is achieved by using a dataset with 10000+ for both fake and real images, and as dataset is large the model will be trained gives more accuracy. Afterwards, Haar cascade classifier gives a remarkable result in feature selection, which makes the CNN work faster and gives the accurate results. Thus, the proposed model gives better accuracy than the existing one.

Table 2: Results of Accuracy attained and loss

Parameter	Accuracy
Training accuracy	97.86%
Test accuracy	98.06%
Training loss	6.17%
Test loss	6.71%

In fig:4 the accuracy of the prediction increases rapidly at the beginning of training, indicating that the network is
learning quickly. Afterwards, there appears to be a flattened off in accuracy which suggests that more epochs are not necessary for it to attain an effective model.

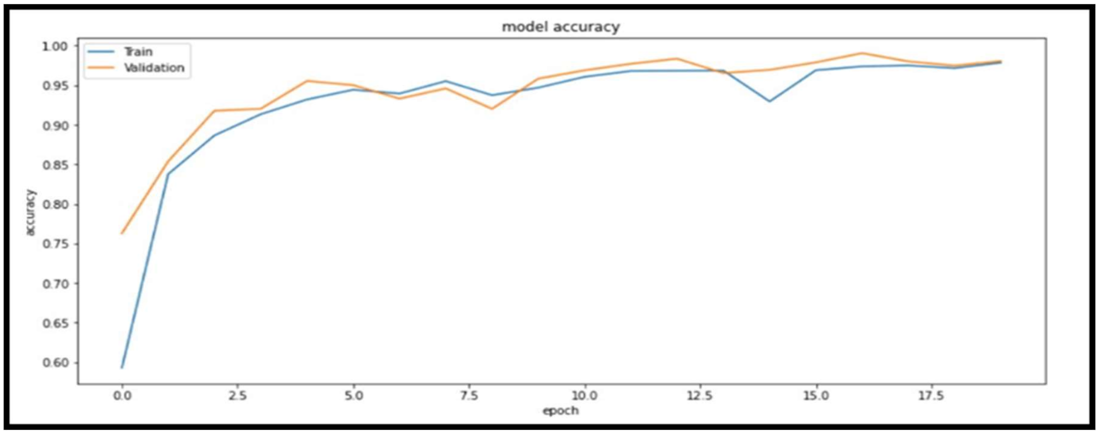


Fig 4. Accuracy vs Epoch

. In fig.5 the loss on the training set decreases rapidly at first, indicating that it is trying to minimize its losses over the training data. This indicates that the optimization process is doing a good job and should be improved upon.

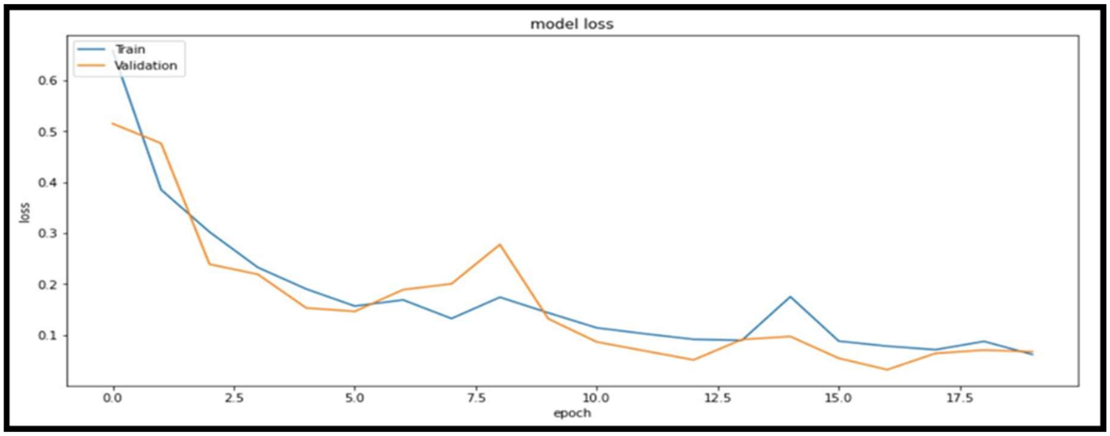


Fig 5. Loss vs Epoch

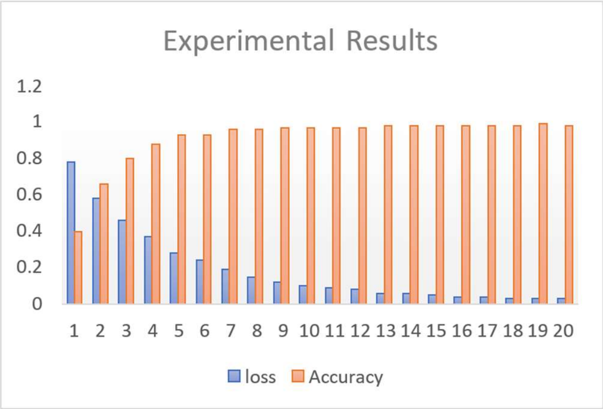


Fig 6: Loss vs Accuracy

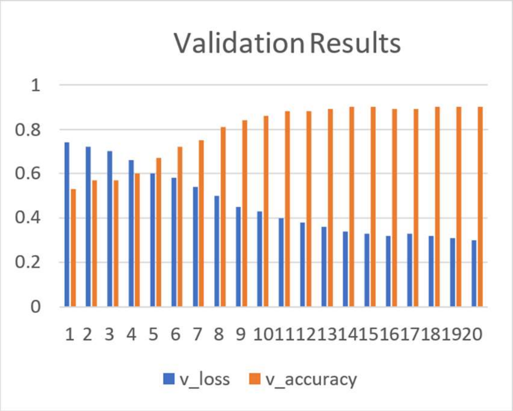


Fig 7: Validation Loss vs Validation Accuracy

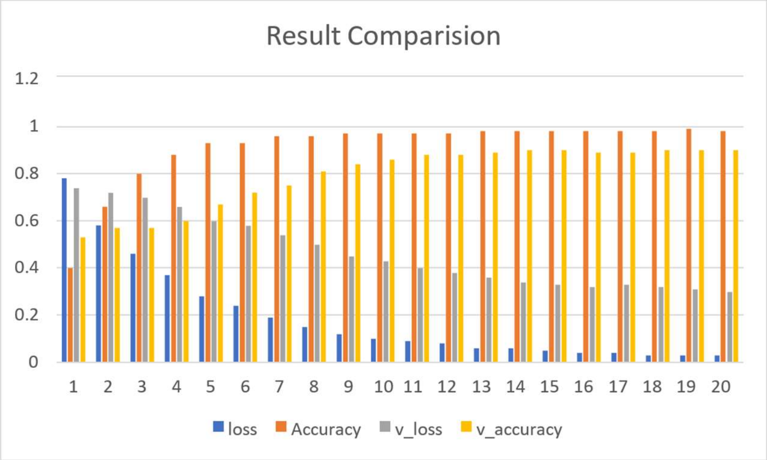


Fig 8: Result compression based on Accuracy and loss

Fig:6 shows the accuracy vs loss. Accuracy score is the number of correct predictions obtained. Loss values are the values indicating the difference from the desired output i.e real or fake. Here cross entropy loss function is used to adjust model weights during training. The aim is to minimize the loss, i.e, the smaller the loss the better the model. A perfect model has a cross-entropy loss of 0 and we reached up to 0.03. Fig:7 shows the validation accuracy vs validation loss. At the beginning of the implementation the dataset is split into training and validation data in the ration of 80:20. Based on the 20% of the dataset the validation results are obtained. Finally, Fig:8 shows a comparison based on experimental results and validation results.

CONCLUSION

A recent advance in technology has allowed images to be manipulated at an accelerated rate. We proposed a system that helps detect real and fake images in this paper. Using the Haar Cascade classifier, we have extracted faces, and then we have used a CNN model to determine if they are real or fake. As the accuracy achieved in the existing system based on the model trained and the dataset used reaches to 0.83. This proposed model gives the accuracy of 0.97 with a minimum loss of 0.67.Thus,the proposed model gives better accuracy than the existing one.

REFERENCES

- MikaWesterlund, The Emergence of Deepfake Technology:A Review.Tecnology Innovation management review,November2019(volume 9 issue 11).
- Afchar, D., Nozick, V., Yamagishi, J., and Echizen, I. (2018, December). MesoNet: a compact facial video forgery detection network. In 2018 IEEE International Workshop on Information Forensics and Security (WIFS) (pp. 1-7). IEEE.
- Sabir, E., Cheng, J., Jaiswal, A., AbdAlmageed, W., Masi, I., and Natarajan, P. (2019). Recurrent convolutional strategies for face manipulation detection in videos. In Proceedings of the IEEE Conferenceon Computer Vision and Pattern Recognition Workshops (pp. 80-87).
- Umakant Dinkar Butkar, Manisha J Waghmare. “ Hybrid Serial-Parallel Linkage Based six degrees of freedom Advanced robotic manipulator.” Computer Integrated Manufacturing Systems, 29(2), 70–82. Retrieved from http://cims-journal.com/index.php/CN/article/view/786
- Li, Y., Chang, M. C., and Lyu, S. (2018, December). In ictu oculi: Exposing AI created fake videos by detecting eye blinking. In 2018 IEEE International Workshop on Information Forensics and Security (WIFS) (pp. 1-7). IEEE.
- Li, Y., and Lyu, S. (2019). Exposing deepfake videos by detecting face warping artifacts. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 46-52).
- Yang, X., Li, Y., and Lyu, S. (2019, May). Exposing deep fakes using inconsistent head poses. In 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 8261-8265).IEEE.
- Mr. Umakant Dinkar Butkar, Manisha J Waghmare. (2023). An Intelligent System Design for Emotion Recognition and Rectification Using Machine Learning. Computer Integrated Manufacturing Systems, 29(2), 32–42. Retrieved from http://cims-journal.com/index.php/CN/article/view/783 .
- Syed Ishfaq Manzoor, Dr Jimmy Singla, Nikita. Fake News Detection Using Machine Learning approaches: A systematic Review.In Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019) IEEE Xplore Part Number: CFP19J32-ART; ISBN: 978-1-5386-9439-8

- Koopman, M., Rodriguez, A. M., and Geradts, Z. (2018). Detection of deepfake video manipulation. In the 20th Irish Machine Vision and Image Processing Conference (IMVIP) (pp. 133-136).
- Jason Bunk, Jawadul H. Bappy, Tajuddin Manhar Mohammed, Lakshmanan Nataraj, ArjunaFlenner, B.S. Manjunath, Shivkumar Chandrasekaran, Amit K. Roy-Chowdhury, and Lawrence Peterson. Detection and Localization of Image Forgeries using Resampling Features and Deep Learning in 2017 IEEE.
- Umakant Dinkar Butkar* & Dr. Nisarg Gandhewar. (2022). AN RESULTS OF DIFFERENT ALGORITHMS FOR ACCIDENT DETECTION USING THE INTERNET OF THINGS. Harbin Gongye Daxue Xuebao/Journal of Harbin Institute of Technology, 54(10), 209–221
- Tewari, A., Zollhoefer, M., Bernard, F., Garrido, P., Kim, H., Perez, P., and Theobalt, C. (2018). High- fidelity monocular face reconstruction based on an unsupervised model-based face autoencoder. IEEE Transactions on Pattern Analysis and Machine Intelligence. DOI: 10.1109/TPAMI.2018.2876842.
- Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X.,Huang, X., and Metaxas, D. N. (2019). StackGAN++: Realistic image synthesis with stacked generative adversarial networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(8), 1947-1962. [15]Lyu, S. (2018, August 29). Detecting deepfake videos in the blink of an eye. Retrieved from http://theconversation.com/detecting-deepfake-videos-in-the-blink-of-an-eye-101072
- [16] Deshmukh, Anushree & Wankhade, Sunil. (2020). Deepfake Detection Approaches Using Deep Learning: A Systematic Review. 10.1007/978-981-15-7421-4_27.
- [17] Anushree Deshmukh and Sunil Wankhade. (2023). An Effective Solution Towards Solving the Problem of Deepfake. Scandinavian Journal of Information Systems, 35(2), 1–8. Retrieved from http://sjisscandinavian-iris.com/index.php/sjis/article/view/557