Video And Image Manipulation Detection Using Multiple Deep Learning Techniques

Sankar Vinayak[1], Alwin Joseph[2], Muhammed Rizal[3]

[1]Student, CSE, Government Engineering College Sreekrishnapuram, Palakkad, Kerala, India
[2]Student, CSE, Government Engineering College Sreekrishnapuram, Palakkad, Kerala, India
[3]Student, CSE, Government Engineering College Sreekrishnapuram, Palakkad, Kerala, India

ABSTRACT

Video and image manipulation, such as deepfakes, and AI-generated and computer-generated media, has become a major concern in recent years due to the advancements in technologies. With the help of these, it is possible for misusing visual media for malicious purposes. Detecting manipulation in videos and images is crucial for various applications, including forensics, journalism, and social media. In this paper, we address the problem of detecting manipulated and generated videos and images using deep learning techniques, with a focus on methods relevant to forensic analysis. We explore and analyze existing methods and technologies for detecting deepfakes and CGI, manipulations, and develop and test novel approaches to arrive at a robust and effective solution. Our evaluation of the performance of different algorithms demonstrates the feasibility and effectiveness of our proposed methods in detecting various types of manipulations. The results of this project have important implications for forensic analysis and other applications where the authenticity of visual media is crucial. The development of reliable methods for detecting manipulations using deep learning techniques can help ensure the credibility and trustworthiness of visual media in various contexts.

Keyword: - *Deep Learning, Image Manipulation Detection, Deepfake Detection, CGI Detection, CNN*

1. Introduction

Video and image manipulation has become a major concern in recent years due to the widespread use of digital media and advances in editing software. The ability to manipulate and generate visual media can have significant impacts on a variety of fields, including forensics, journalism, and social media. Image alteration is now simpler than ever because of the recent growth of digital photos and the widespread use of gadgets like smartphones and tablets. Furthermore, the public now has even greater access to this activity thanks to the low cost of picture modification software. As a result, some photographs are altered so skillfully that even humans are unable to tell the difference. As such, detecting manipulation in videos and images has become an important research topic. Existing methods for detecting manipulation in videos and images include traditional forensic analysis techniques, such as error-level analysis and image quality assessment, as well as newer approaches using machine learning and computer vision. Deep understanding is effective for detecting manipulation in images and videos due to its ability to learn features and patterns from data.

2. Proposed system

Our proposed system utilizes multiple deep-learning-based models to effectively address various types of manipulations. The complete functioning of the model and its training process are detailed in Figure 1. To detect Deepfake videos, manipulated images, and generated images, different techniques will be employed. Data for training and testing will be gathered from a combination of standard datasets and publicly available files on the internet. To ensure user-friendliness, a straightforward user interface will be provided, allowing easy utilization of the different models by end users. The preprocessing of files before feeding them into the respective models will be seamlessly managed by the user interface.

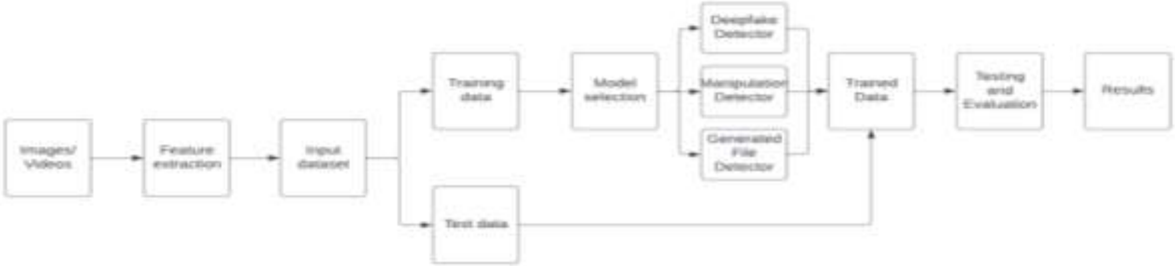


Fig-1: Block diagram of system

3. Methodology

We use a total of three different deep learning models for different manipulations namely Deepfake Detector, Synthetic Detector, and Image Manipulation Detector

3.1 Deepfake Detector

The deepfake detection model network consists of three main components, a convolutional neural network (CNN) architecture for feature extraction, a long short-term memory [1] (LSTM) network for temporal modeling and a fully connected layer for classification. The CNN used in this architecture is the ResNext50[2] 32x4d model, which is pre-trained on the ImageNet dataset and has been shown to perform well on various computer vision tasks. The

pre-trained weights are used for the CNN layers. The output of the CNN is passed through an adaptive average pooling layer that reduces the spatial dimensions to 1x1 and averages the feature maps across all spatial locations. This results in a fixed-length feature vector that captures the salient features of the input frames. The feature vector is then fed into an LSTM network which is responsible for modeling the temporal dynamics of the video.

The output of the LSTM is passed through a dropout layer to prevent overfitting and then fed into a fully connected layer. The output of the fully connected layer is compared to the ground-truth labels to compute the loss.

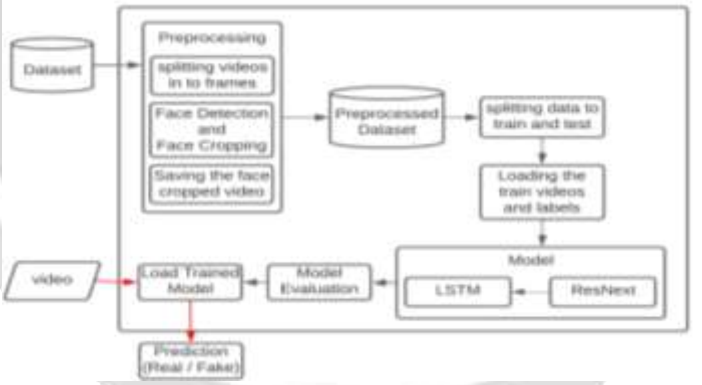

Dataset
Preprocessing
splitting videos in to frames
Face Detection and Face Cropping
Saving the face cropped video
Preprocessed Dataset
splitting data to train and test
Loading the train videos and labels
Model
LSTM
ResNext
Model Evaluation
Load Trained Model
video
Prediction (Real / Fake)

Fig-2: Architecture of Deepfake Detection Model

3.2 Synthetic Detector

The method we used here is based on the Vision Transformer [3]. It is a transformer-based architecture used in image classification and speech recognition applications. The weights of ViT are based on the values available after training it on the ImageNet dataset. To reduce the computational requirements the image gets resized into 224 × 224 before passing into the network. Also, the pixel value’s mean and standard deviation are shifted which helps in better generalization and reducing computation. The architecture of ViT contains many stages

In the Vision Transformer (ViT) architecture, the input image is first divided into a sequence of image patches, and a position number is assigned to each patch to indicate its order. This patch sequence, along with the position embeddings, serves as the input to the subsequent layers. The embedded patches undergo a linear projection, where each patch is transformed into an embedding. Using an embedding instead of the raw image values allows for a learnable representation of the image, which can improve performance through training. Layer normalization is applied to regularize the neural network and reduce overfitting. This technique ensures that the activations of the network remain stable and consistent across different examples. The architecture includes a multi-head attention layer, which performs self-attention on the embedded patches. Self-attention allows the model to focus on different parts of the image when processing each patch, capturing meaningful relationships between patches and enhancing the model's understanding of the image. A multi-layer perceptron (MLP) block, consisting of multiple feedforward layers, is utilized to further process the output of the attention layer. The MLP block helps in capturing complex patterns and refining the features extracted from the patches. The transformer encoder is the combination of all the aforementioned layers. It takes the embedded patches, applies multi-head attention, processes them through an MLP block, and repeats this process for multiple encoder layers. This series of layers enables the model to capture intricate image features and learn representations that can be utilized for various tasks.

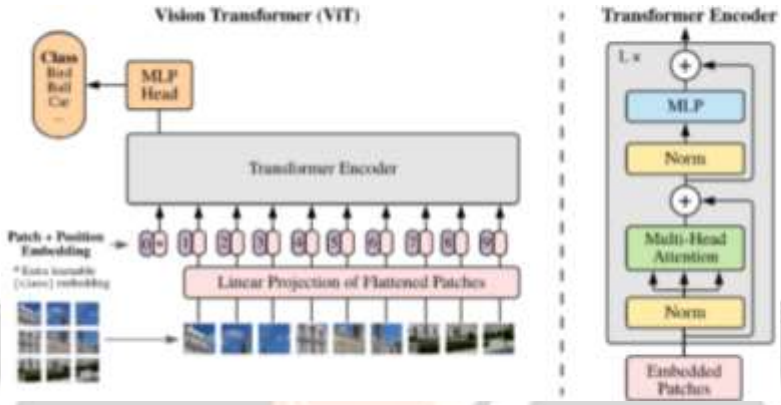


Fig-3: ViT architecture

The MLP Head consists of layers which take the input from the ViT output and convert it to lower dimensional output before passing through the non-linear activation function. Here the output from ViT which is of shape 768 is reduced to 256. It is then converted to the required output dimension of three, all by using a fully connected feedforward neural network. The non-linear activation before the output layer, in our case, rectified linear unit, which is used for improving the prediction confidence and eliminating the connections which do not affect the results. All the weights values except for the MLP head are frozen during the training process and the MLP head is trained to fit the requirement for improving the performance of the model an optimal base learning rate is set as 0.001 and the batch size for training the model is set to 64. The learning rate was set to reduce by 0.5 if the loss does not improve over three epochs.

3.3 Image Manipulation Detector

The main objective of Image Manipulation Model is to develop a pipeline that can differentiate between manipulated and genuine images. To achieve this goal, inspiration was drawn from Y. Rao et al’s [4] proposed architecture, which employs a CNN as a feature extractor that takes an image patch $X \varepsilon R^{p \times p}$ as input and generates a feature representation $Y = f(X) \varepsilon R^{\kappa}$, where K represents the number of dimensions. The resulting feature representation is then processed by an SVM classifier to determine whether the input image is authentic or manipulated. CNNs, which are members of the deep neural network family, are mostly used for image processing. Multiple convolutional layers, fully connected layers, and a softmax classifier make up the basic structure of a CNN. The input and output of each convolutional layer are referred to as feature maps. A convolutional layer is made up of a convolution, a non-linear activation function, and a pooling operation.

The CNN utilized (**Fig-4)** comprises of nine convolutional layers and two maxpooling layers, with the input size of the network being a 128x128x3 patch, where 3 indicates the RGB colour channels. The first two convolutions have a kernel size of 5x5 and output 3 and 30 kernels, followed by a pooling operation using a 2x2 filter. The subsequent eight layers have 16 kernels, with convolutions and max pooling using 3x3 kernels and 2x2 filters. All convolutional layers use the ReLU activation function, and local response normalization is applied to every feature map before the pooling operation to enhance generalization [4]

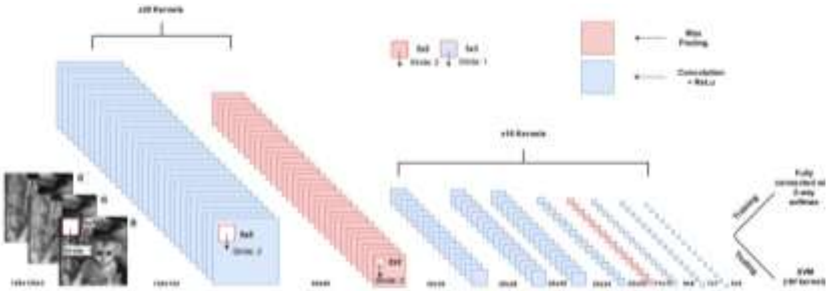


Fig-4: CNN architecture

To train the CNN, image patches with a size of 128x128x3 are extracted from the dataset with the help of a mask to give the CNN architecture the capacity to focus on the local areas of the artefacts and learn to detect them. The patches are extracted using the aforementioned method and then entered into the CNN. A 400-dimensional (5x5x16) feature representation of the patches is created by the CNN and fed into a fully-connected layer afterwards. This layer employs a dropout-based 2-way softmax classifier [5]. The SVM classifier uses this feature vector to differentiate between genuine and altered images after the CNN extracts the 400-dimensional feature representation of the image patches. The pipeline is implemented in Python 3.10 using PyTorch 2.0 to implement the CNN and the SVM3 [6] implementation provided by sci-kit learn.

4. Training and Result

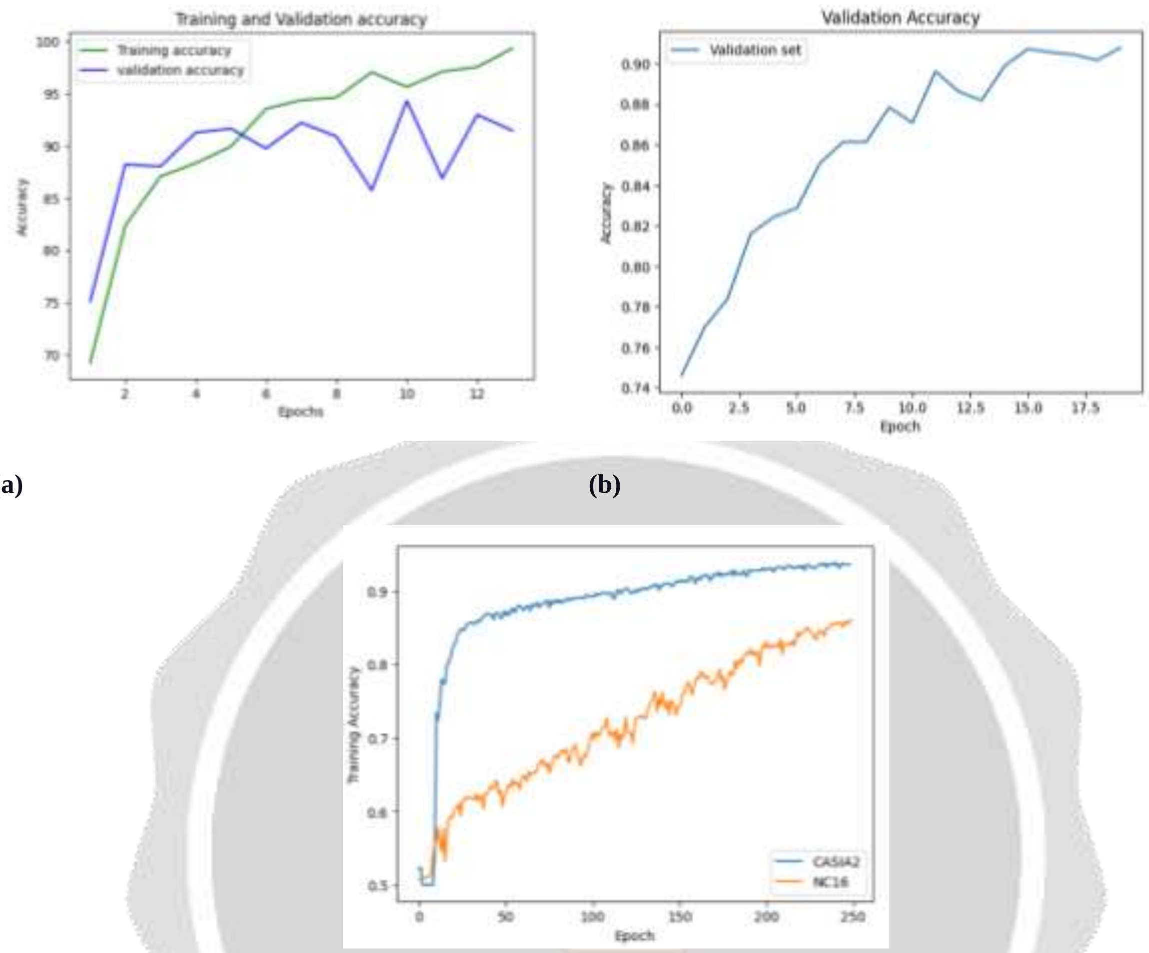
We have developed an application that integrates various specialized models to discern genuine and fabricated images and videos. The different models were trained using different datasets which are explained below. The application features a user-friendly interface framework, simplifying the creation process while hiding the intricate inner workings.

4.1 Deepfake Detector

The deepfake detector model was trained and tested using the Celeb-DF [7] dataset. To create a face-only dataset, a subset of the Celeb-DF dataset was used. The dataset was divided into 80% for training and 20% for testing. The training set consisted of 657 real videos and 920 fake videos, totaling 1577 videos. The test set contained 232 real videos and 294 fake videos, amounting to 526 videos in total. The model was trained with a batch size of 4 and a learning rate of 1e-5 (0.00001) over 13 epochs. The training utilized the cross-entropy loss function, and the Adam optimizer with a weight decay of 1e-3 was employed for parameter optimization. The model's performance was evaluated using the accuracy metric on a separate test set. After 13 epochs, the model achieved an accuracy of 91.44% and a loss of 0.26. However, the model did not demonstrate further improvement, and its performance began to diverge afterwards.

4.2 Synthetic Detector

Our proposed synthetic detector model outperformed the Efficient-Net [8] classifier, another image classification model, in terms of performance. While our model's generalization capability may diminish with smaller batch sizes and dataset sizes, the pre-training of the base model facilitated strong performance from the initial epoch. Accuracy and loss values demonstrated comparable performance on both the training and testing sets. As training progressed, the model's performance improvements became incremental, and the learning rate was automatically adjusted for smaller step sizes to enable gradual progress. Around 20 epochs, the performance on the train and test sets started to diverge, prompting the implementation of early stopping to ensure the model's optimal state during the epoch. The final model achieved approximately 91% accuracy, exhibiting good precision and recall abilities on the validation set. A visual representation of these metrics can be observed in the provided figure. The model gradually reached a plateau, and continuing training resulted in either negligible gains or no improvement at all.



(a) **(b)**

(c)

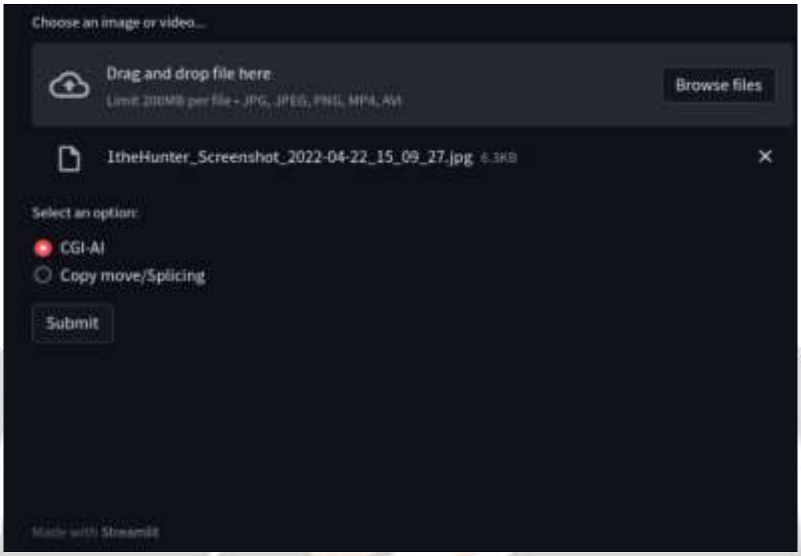
Fig-5: Accuracies of **(a)** Deepfake Detection Model, **(b)** Synthetic Detection Model, **(c)** Image Manipulation Detection Model

4.3 Image Manipulation Detector

The image manipulation detector model was trained and tested using the CASIA2[9] and NC16[10] datasets. Patches were extracted from 2,204 images of the NC16 dataset and 20,076 images of the CASIA2 dataset without data augmentation. The CNN model achieved an accuracy of 82.3% after 30 epochs, as depicted in the figure. Subsequently, the accuracy improvement became gradual, ultimately reaching 93.5% after 250 epochs. The model's loss steadily decreased during the training period, converging to a final value of 0.365, as shown in the figure. SRM initialization filters were applied in the first convolutional layer of the CNN, and the hyperparameters of the CNN were individually optimized for each dataset. The CNN-generated features were combined using mean feature fusion, and an SVM model with an RBF kernel was employed for classification. The SVM model's accuracy was assessed using 10-fold cross-validation, resulting in an accuracy of 0.9682 and an error rate of 0.0119. Additionally, the evaluation reported 1426 true negatives, 72 false negatives, 17 false positives, and 1008 true positives.

5. Output

The models are combined and deployed as a web app using Streamlit[11] framework. Our designed user interface provides the user with the ability to upload image and video files and select different options for detecting various forms of manipulation. The user is presented with a file uploader widget that enables them to select an image or video file. Depending on the type of file uploaded, relevant options are displayed on the radio buttons. The user can select an option by clicking on the corresponding radio button. A visual representation of the output can be observed in **Fig-6** which is shown below.



Choose an image or video...

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG, MP4, AVI

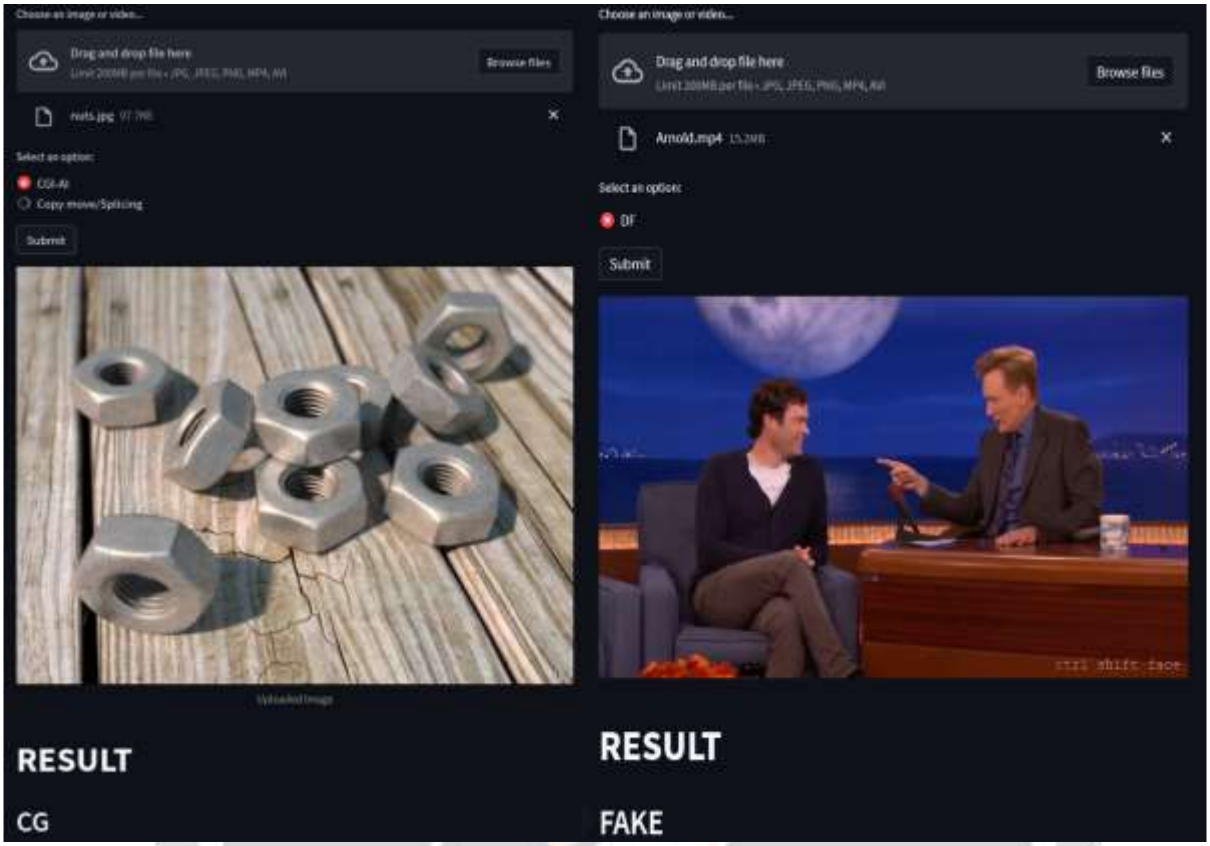
ItheHunter_Screenshot_2022-04-22_15_09_27.jpg 6.3KB

Select an option:

● CGI-AI

○ Copy move/Splicing

(a)


Choose an image or video...
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG, MP4, AVI
Browse files
Uploaded Image
Choose an image or video...
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG, MP4, AVI
Browse files
Submit
Submit
ctrl shift face

nuts.jpg 97.7KB

Select an option:

● CGI-AI
○ Copy-move/Splicing

RESULT

CG

Arnold.mp4 15.2MB

Select an option:

● DF

RESULT

FAKE

(b) (c)

Fig-6: Result screenshot (a)UI, (b)Image input, (c) Video input

6. CONCLUSIONS

The "Video and Image Manipulation Detection Using Multiple Deep Learning Models" project focuses on utilizing deep learning techniques to address the growing concern of deepfakes, AI-generated, and computer-generated media, which pose serious threats due to their potential misuse. These manipulated media can be used to spread false information or harm individuals' reputations. The primary objective of the project is to develop reliable methods for detecting manipulations in visual media, with a strong emphasis on forensic analysis. The project involves a comprehensive exploration and analysis of existing technologies for detecting deepfakes and CGI manipulations. Additionally, novel approaches are developed and rigorously tested to establish a robust and effective solution.

Through performance evaluation, the project demonstrates the efficacy of the proposed methods in detecting various types of manipulations. These methods have significant applications in forensic analysis, journalism, social media, and other domains where the authenticity of visual media is paramount. By ensuring the credibility and trustworthiness of visual media, these techniques can assist in investigations and legal proceedings. In conclusion, the "Video and Image Manipulation Detection" project represents a valuable contribution in combating the escalating issue of manipulated visual media. The proposed deep learning techniques show promise in detecting manipulations and hold the potential to make a substantial impact in contexts where the authenticity of visual media is critical.

6. REFERENCES

[1]. Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). “Long Short-term Memory. Neural computation”. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[2]. S. Xie, R. Girshick, P. Dollár, Z. Tu and K. He, “Aggregated Residual Transformations for Deep Neural Networks”, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 5987-5995, doi: 10.1109/CVPR.2017.634.

[3]. Dosovitskiy, Alexey, et al. “An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale.” ArXiv, 2020, /abs/2010.11929.

[4]. Yuan Rao and Jiangqun Ni. “A deep learning approach to detection of splicing and copy-move forgeries in images”. In 2016 IEEE International Workshop on Information Forensics and Security (WIFS), pages 1-6. IEEE, 2016.

[5]. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In Advances in neural information processing systems, pages 1097-1105, 2012.

[6]. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[7]. Y. Li, X. Yang, P. Sun, H. Qi and S. Lyu, “Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics”, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 3204-3213, doi: 10.1109/CVPR42600.2020.00327.

[8]. Tan, Mingxing & Le, Quoc. (2019). “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”.

[9]. J. Dong, W. Wang and T. Tan, “CASIA Image Tampering Detection Evaluation Database”, 2013 IEEE China Summit and International Conference on Signal and Information Processing, Beijing, China, 2013, pp. 422-426, doi: 10.1109/ChinaSIP.2013.6625374.

[10]. Open Media Forensics Challenge (OpenMFC) Evaluation Datasets. https://www.nist.gov/itl/iad/mig/open-media-forensics-challenge

[11]. https://streamlit.io/

[12]. Ferreira, S.; Antunes, M.; Correia, M.E. “Exposing Manipulated Photos and Videos in Digital Forensics Analysis”. J. Imaging 2021, 7, 102. https://doi.org/10.3390/jimaging7070102

[13]. D. Güera and E. J. Delp, “Deepfake Video Detection Using Recurrent Neural Networks,” 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2018, pp. 1-6, doi: 10.1109/AVSS.2018.8639163.