



Sri Lanka Institute of Information Technology

Technical Paper

Project – Deep Fake Detection Sytem

Student Name	Student ID
Edirisinghege E M N	IT21340864
Anupama K G A	IT21345678

Table of Contents

Abstract	3
Problem Statement	3
Objectives	3
Product Scope	3
Methodology	4
Collection of data	4
Requirements and Analysis.....	6
Software & Hardware Requirements	6
Functionality Requirements	7
Design	7
Client Side.....	7
Server Side	7
Implementation	8
Client Side.....	8
Server Side	8
Tools & Libraries	8
Testing.....	8
Evolution.....	9
Dataset.....	9
Model Performance.....	9
Cross-Validation	10
Qualitative Analysis.....	10
Comparative Analysis:.....	10
Deployment Testing.....	10
Future Works	10
Conclusion	11

Abstract

There are now significant concerns about the accuracy of digital information because of the development of deepfake images produced by complex algorithms caused by the massive development of artificial intelligence (AI) and machine learning (ML). In this project, a reliable method for detecting deepfake photos is researched, designed, and put into practice. Deepfakes are successfully identified by the built model, which improves security measures, by using deep learning techniques and a carefully selected set of authentic and modified pictures. The strategy, implementation, assessment, and contributions to the area of secure software development are covered in this project.

Problem Statement

Deepfakes Dataset tends to spread quickly, so it would be hard to differentiate on the basis of their photos real and unreal. These systems have to be very accurate or else there could be a lack of trust in the internet by creating skeptical and false thoughts. It is advised to use the latest technology in the surveillance system due to improvement in forging methods that result in deepfake activities.

Objectives

- Develop the deep learning and machine learning based system to detect Deep fakes
- Ensure the users privacy and confidentiality
- create the user friendly interface to operate the functionalities and analysis
- Ensure fast response and higher accuracy for the deep fake predictions
- When the deep fake detected the system notify the alert and the identified image save as the log
- Intergrade with the social media

Product Scope

- **Compliance & Regulation** – Supports compliance, law, regulation, and governance by producing detailed reports on deep fake detection efforts and quick response mechanisms
- **User Login** – Provides facial recognition api to user login sytem
- **Sensitive data protection**-Increase users privacy and protect sensitive data using machine learning to detect manipulated media content.
- **Cost effectiveness** - For the purpose of predicting future threats related the deep fakes, machine learning is used, and the highest quality predictions at the industry level are taken into consideration, Other deep fake detection tools are more price and large size but but this tool more lightweight and cost efficient.

- **User interface** - Developing a colorfully user-friendly user interface for deep fake detection and alerting. It has a clear simple approach to notify and setup the deep fakes.
- **Higher accuracy** - Improve deep fake detection accuracy by using deep learning and advanced machine learning models capable of deep fake media.

Methodology

This project main focus identifies and discovering insider threats in the organization using machine learning.

The system procedure goes as follows:

- Collection of data
- Data pre-processing
- Data Analytics using ML
- Analyzing the Result

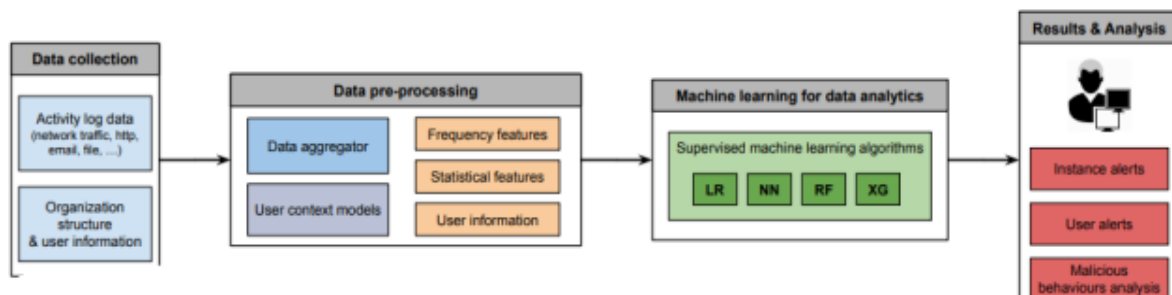




Figure 1 Methodology

Collection of data

Dataset selection is a most important part of the deep fake detection. Because the Obtaining source images from trustworthy providers ranging from real to photo shopped with diverse illumination conditions and styles of modification used were the objectives of the selection process. The size and grade of the data set are the main pillars on which the training of a best-in-class detection model capable of generalization to different real-world situations are founded. [1]


MANJIL KARKI · UPDATED 2 YEARS AGO

42
New Notebook
Download (2 GB)



deepfake and real images
Detect if any images is real image of deepfake image

Data Card Code (65) Discussion (0) Suggestions (0)


About Dataset

This dataset contains manipulated images and real images. The manipulated images are the faces which are created by various means. The source for this dataset was <https://zenodo.org/record/5528418#.YpdIS2hBzDd>

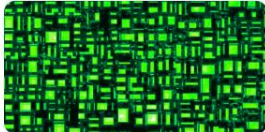
this dataset was processed as our will to get maximum outcome out of these images. Each image is a 256 X 256 jpg image of human face either real or fake

Usability 6.25
License Unknown
Expected update frequency Never
Tags
Image Social Networks
Research
People and Society
Machine Learning Ethics

Figure 2- deepfake and real images dataset


YI HAO PUAH · UPDATED 3 YEARS AGO

7
New Notebook
Download (2 GB)



Deep Fake Images

Data Card Code (1) Discussion (0) Suggestions (0)

About Dataset

This dataset was used for the SeeTrue hackathon and is another popular dataset in the DeepFake detection challenge.

Usability 2.94
License Unknown
Expected update frequency Not specified
Tags

test (2 directories, 3 files)




Figure 3 - dataset 2

a. Data pre-processing

In this process approach applies a multi-step data preprocessing strategy.Obtaining and preprocessing a large number of data sets for training and testing models [1]

b. Data Analytics using ML

Analyze the preprocessed data using various types of machine learning algorithms and methodologies to identify patterns and structures. This is a critical part of the project so that by using machine learning we can train modules that can manipulated image and any other media content. Various algorithms such as decision trees, Logistic Regression (LR), Neural Network (NN), Random Forest (RF), and XGBoost (XG) will be explored to identify the most appropriate ML model. [2]

c. Analyzing the Result

Conduct the testing phase to evaluate system performance. Accuracy, F1 score, AUC ROC and many other metrics are used for this. It also analyzes identified which uploaded images are real or fake. [3]

Requirements and Analysis

The deepfake detection system's performance depends on a detailed examination of the project specifications and a thorough assessment of the resources and limitations at reach. This section describes the procedures used to determine the needs and examine the project environment.

Software & Hardware Requirements

Thus, each step towards a deep fake detection system will be guided by the proper identification of hardware and software requirements. The hardware needs comprise high-performing CPU or GPUs that are capable to take on heavy computational roles, especially, during attentions model training and inference. Software specifics consist of programming languages, base libraries, and frameworks used for the units of system features. As for me, I use widely-used tools, like Python, TensorFlow, Keras, opencv and Flask, as they can be adjusted to my assignments and have an active community. [4]



Figure 4

Functionality Requirements

- **Browse and select image** - User must be able to select images which are used for predicting deep fake or not using user friendly fast response UI.
- **Preprocessing and training model** - System must be able to process large number of data and training model using scaling normalization and more mechanism.
- **Deep fake Detection** - The system should detect accurately which uploaded image is real or fake and when it is fake display an alert immediately.
- **Logging and Reporting** - System must be able to be tracking and saving the history and user activity of these tool usage.

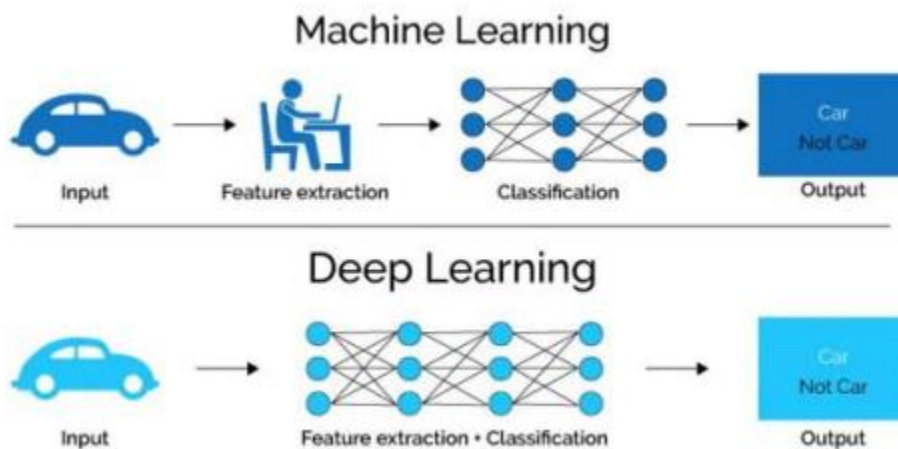


Figure 5

Design

Client Side

Design interactive powerful User interface to user using HTML, CSS and JavaScript. Addiotnaly, the client side interface provide fast response and higher accuracy result.

Server Side

Deep learning model is made up by Python and relies on TensorFlow and Keras along with server side. The given architecture model acquires its structure from the use of CNNs with several layers aimed at feature extraction as well as classification. Through of transfer seaching algorithms the pre-trained models are utilized to accelerate the training speed. [5]

The deployed server-side component incorporates the lightweight Flask framework for Python in its infrastructure. Flask integrates the deep learning model as a web service for the implementation of client applications, so that interaction becomes smooth. Through Flask, the

framework provides overcomes scalability or some accessing issues while it delivers user-friendly interface for managing the features of deep fake detection.

Additionally, Using transfer are utilized to take advantages of previously prepared architectures of the CNN (VGG16 or ResNet) for extension features of the image. Fine tuning is used only on the last layer of the network and for the detection of this very specific task.

Implementation

Client Side

- **Visual Studio Code:** Powerful code editor used for developing the HTML, CSS, JavaScript client applications to interactive user interface
- **HTML, CSS, JavaScript:** Technologies used for designing and enhancing the user interface, ensuring efficient handling of user inputs and communication with the server and providing a seamless user experience.

Server Side

- **Python:** Main programming languages of the project that used for activities of machine learning, deep learning and user account management.
- **Flask:** A lightweight Python web framework, an excellent approach of deploying the deep learning model as a web service. Flask acts as an HTTP handler, it sends the frontend application to the user and lets him interact with the model of the deepfake, which is then trained.

Tools & Libraries

- **Numpy** - which is optimization of the processing and storage support for large, multi-dimensional arrays and matrices, including a substantial collection of high performing mathematical functions to operate on these arrays.
- **OpenCV** - Image processing tool that library that can be used to perform tasks like face recognition, objection tracking and more.

Testing

Unit Testing - Individual component by component testing process. Components contain network monitoring tool, deep fake detection tool, malware detection tool. Individual compaction by compaction testing process. The components contain network monitoring tool, deep fake detection tool, malware detection tool. This testing process checks that individual related functions and functions work correctly and accurately. [6]

Integration Testing- Tested after integration between frontend and backend Python scripts and flask. Its ensures seamless application functionality and activities.

System Testing – The software was tested to run with less hardware usage. It ensures the software is more optimized, fast and accurate

Evolution

The evaluation of the deepfake detection system is done by a complete analysis of the sought features involved through the vital metrics and hence, the wellness of the model for detecting the changed images is delivered. Evaluation takes place through the use of quantitative and qualitative parameters simultaneously.

Dataset

- **Training and Testing Data:** Data set has all brand of real images and deepfake quite possibly borrowing others' to face databases such as FaceForensics++, DeepFake Detection Challenge and Celeb-DF.

Metrics

- **Accuracy:** The model's accuracy is evaluated in both general cases and counted number of fakes, and the pictures that are true, as well as the picture total altogether.
- **Precision:** The assessment cannot be based on the ratio of the true detection to all positives but can be made and no false positives occur.
- **Recall:** To estimate the fraction calculates how many real positive examples are predicted as Altered by The deep fake detection deep fake detection Model.
- **F1 Score:** Handles also this challenge by balancing faithfulness and high level of safety for both of these metrics. [7]
- **AUC-ROC (Area Under the Receiver Operating Characteristic Curve):** AUC represents the model's skill to distinguish between different classes which is expressed by the AUC and the higher fallout of AUC the better evaluated . [8]

Model Performance

- **Training and Validation Loss:** The monitor is implemented for the training process to make sure that the model not only found the topics effectively be able to also make new patterns from the training data.
- **Confusion Matrix:** Discusses how alike are the counts of the true positive, true negative, false positive and false negative rate. It also gives a full meaningful analysis of the current status.

Cross-Validation

- **K-Fold Cross-Validation:** The tool is based on evaluation of the performance of the model through the use of K-fold cross-validation method where training subset is matched with validation subset for each instance of which sample size equal to K.

Qualitative Analysis

- **Visual Inspection:** Comparison of these type decisions from our model on a given image which will provide us with an insight about any patterns that might be associated with the misclassification, and thereby what needs to be updated
- **Case Studies:** In-depth examination of individual cases is used as a basis for providing evidence of both the strengths and weaknesses of the philosophy.

Comparative Analysis:

- **Benchmarking:** Evaluating the models performance vs. any other models for deepfake detection by computing performance based on newly released standardized datasets .

Deployment Testing:

- **Load Testing:** The model of operations has been checked on reacting to varying load settings while running them with the desired performance level.
- **Latency and Throughput:** While determining the segment of time or detecting whether it is the throughput is okay or which has to be increased within a reasonable time, the system should work best in conditions of the real-time as well as in the near-real-time.

Future Works

Enhanced Model Architecture: Increase the amount of the deep learning model's architecture by reading different complex approaches such as attention mechanisms, capsule networks, transformer models, and so on. In order to find out complicated ensembles methods that consist of more models together for better performance, keep exploring.

Adversarial Training: Use methods of model adversarial training that are most effective to end up with the security model which restricts the most at adversarial attacks. Ensembles is one of the methods of training, and adversarially perturbed data which is used by the model to remain sturdy in the face of tampering is one of the approaches that can be applied.

Active Learning: Sign up for efficient active learning mode of the training sets that acknowledge the scenario for judicious training/labeling of the samples. It will be more potent of a tool with an increasing emphasis in a case where the limited number of labeled resources are used. Also, it will achieve a positive result with lesser amounts of data labelled.

Multimodal Detection: Emphasize enhancement of surveillance abilities by increasing capacity to capture both image and also audios, videos, and text. Subsequently, enmeshing multi-layered detection methods would not just for the sake of securing against deepfake content, but would also ensure that such measures are duly available across all online platforms.

Real-Time Detection: Establish effective mechanism to detect deepfakes early for immediate identification and responding to any deepfake-like image or video uploaded online. In other cases, it could be linked with the model creation (low-level services for edge devices) or with high scalability (cloud services).

User Feedback Integration: While the model is being developed, give users a feedback form to them to provide their response to the encountered deepfakes in order to enhance the model and minimize the possibility of false positives/negatives. Integrate the machine learning function into system software so that it analyzes users' comments and makes upgrade if necessary.

Privacy Preservation Techniques: Explore the ways which guarantee the confidentiality and the detection of the deepfakes without users of the nervous distraction. This can be done for instance by any of say differential privacy method, federated learning method, or secure multi-party computation method.

Conclusion

In our project on the latter by designing a robust deepfake detection system, and thus to the danger of deepfake spread. By using machine learning and deep learning, we created the tool that can find the manipulated media just as accurately, but it took us into consideration too user privacy. Our system's easy to use interface design, adherence to various rules as well as social media integration ensures its usability and influence. In the future, expect the growth of new models, methods of training models and relationships with people who are able to stop the spread of deepfakes that will lead to the strengthening of the possibilities to protect the integrity of digital information.

References

- [1] .. D. C. W. I. i. Khalid K. Al-jabery, "Data Preprocessing," no. 2020.
- [2] S. Yıldırım, "15 Must-Know Machine Learning Algorithms," [Online]. Available: <https://towardsdatascience.com/15-must-know-machine-learning-algorithms-44faf6bc758e>.

- [3] M. Saini, "Understanding Performance Metrics in Machine Learning: Precision, Recall, F1 Score, Confusion Matrix, and AUC ROC with an example," [Online]. Available: <https://www.linkedin.com/pulse/understanding-performance-metrics-machine-learning-precision-saini#:~:text=In%20conclusion%2C%20precision%2C%20recall%2C,our%20models%20for%20better%20results..>
- [4] E. F. Deisy Chaves, "CPU vs GPU performance of deep learning based face detectors using resized images in forensic applications," no. 2019.
- [5] N. S. Ashifur Rahman, "Deepfake Video Detection Using CNN and RCNN This project "Deepfake Video Detection Using CNN and RCNN" report submitted by Ashifur," no. 2022.
- [6] geeksforgeeks, "Unit Testing – Software Testing," 2024. [Online]. Available: <https://www.geeksforgeeks.org/unit-testing-software-testing/>.
- [7] encord, "F1 Score in Machine Learning," 2023. [Online]. Available: <https://encord.com/blog/f1-score-in-machine-learning/>.
- [8] A. Bhandari, "Guide to AUC ROC Curve in Machine Learning : What Is Specificity?," 2024. [Online]. Available: <https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/#:~:text=The%20Area%20Under%20the%20Curve,the%20positive%20and%20negative%20classes..>
- [9] M. C. T. R. F. T. J. R. S. J. W. C. M. L. Collins, "Common sense guide to mitigating insider threats, fifth edition," 2016.
- [1] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2019.