Real-Time Deepfake Image Detection Using Pretrained Xception CNN

Muhammad Asif P A  
Department of Computer Science EngineeringRajalakshmi Engineering CollegeChennai , India  
230701199@rajalakshmi.edu.in

Rakesh R  
Department of Computer Science Engineering  
Rajalakshmi Engineering CollegeChennai , India  
230701257@rajalakshmi.edu.in

R Bhuvaneswari  
Department of Computer Science and Engineering  
Rajalakshmi Engineering CollegeChennai , India   
bhuvanacheran@gmail.com

*Abstract*—**Deepfake tech now growing too fast, and it's becoming a big problem for digital security, privacy, and also spreading fake info. This project gives one proper system to catch deepfakes by using one strong pretrained deep learning model, which is the Xception Convolutional Neural Network (CNN). In this system, users just upload the image, then the system will use this Xception model to check if it’s real or fake. If it finds the image is fake or edited, it will give the user an option to directly report it to cyber security people for further action. The platform is simple to use, with clean interface in frontend. Backend side, it handles image checking and also user login using Firebase. Since we used Xception model which is already trained and powerful, it gives very good performance without needing too much processing power. The model is strong because of transfer learning, and can detect deepfakes properly in different situations.As fake image tricks are becoming smarter, this kind of AI tool is very useful to protect online platforms, user identity and even help in legal matters. In future, we will try to improve this system more, to make it work better even when someone tries to fool it with advanced methods. Also, we will increase the data so it can catch more types of fake edits.**

**Keywords— Deepfake detection, Xception CNN, transfer learning, image checking, fast detection, Firebase login, protection from fake images, cyber**

# Introduction

As AI is growing very fast now, deepfake tech also become one big headache for online safety and digital trust. These fake but real-looking images and videos are made using high-level machine learning, mainly using GANs (Generative Adversarial Networks), which can create content that looks totally real. Because of this realistic look, deepfakes can fool people easily, and this is now a big problem — spreading fake news, identity theft, cyber crimes and all. This serious issue now shows that we need some strong tool to catch this fake content automatically.

Now deepfake making tools are very easy to use, anyone can access them — which means the risk of fake news and cheating is becoming bigger. People can change politician speeches, fake celebrity videos, or even legal papers — this is dangerous. So our project gives one good solution — we used the Xception CNN model, which is very good in image classification. We used transfer learning here, so it already knows a lot, and can work even on low-power systems or with less data.

In this system, users can upload any image to check if it’s fake or not. If the model finds it’s a manipulated image, user can report it directly to cyber security team. This way, users stay alert and also help in legal action. The whole system is made user-friendly. Frontend is done using React.js for smooth working. Backend is made using Flask or FastAPI to process images quickly. Also, Firebase is used for secure login and data handling — so both normal people and organizations can use this system. This project also studied other deepfake detection methods — checked their pros and cons. But our system stands out because it uses a powerful CNN model with a working, easy-to-use platform. This tool can be useful in many places — news/media checking, police/cyber team, online education, and even government works.

# **Literature Survey**

Deep a content is usually created on purpose often spread in social medias to create mistrust. To identify deepfakes , experts uses several technologies and clues to identify whether it is real or fake. This is done by Classifying Images , Data Augmentation , Transfer Learning. The first method classifying images is done by teaching the computers to spot difference between real and fake images by looking specific details. The second method is data augmentation it is done by adding more examples to help the computer learn better. The last method it is transfer learning is done by using a pre trained model that is already good that spotting patterns , it can be fine tuned further for more specific image detection. One of the best tool that is the Xception model which is really good at finding the clues because it's been trained to notice small deviation in images. By combining this with transfer learning and regularization it can able to detect deepfakes with 93.01% accuracy [1]. Deep learning is like a smart computer brain that is great at spotting fake news which is better than older methods. This uses tools like CNNs , RNNs, BERT, which are different types of learning methods which find patterns in data. Word embeddings, feature extraction, natural language processing these models helps to get better when they use multimodel inputs which means combining different types of information like text, imagery , contextual information. [2]. Detecting fake videos is difficult than the images but computers can use machine learning techniques to solve this problem. Filtering, feature extraction, classification. Study show that checking longer parts of a video helps the computer to give a more accurate prediction [3].

Deepfakes like fake news is created intentionally, usually spreads in social medias often leading to heated debates. To detect this the computers look for abnormal or clues, auspicious image or video, clickbait headings. We use a of smart tools called hybrid architecture like CNNs , RNNs and GNNs. The system tracks how quickly is spread. [4]. Deepfake images are ai made fake image that look real. To detect this computer use Siamese 3D CNNs with spatiotemporal attention, this compares images and mainly focus on key details like shapes and motions. They also check texture features like GLCM, LBP and SLP [5]. Plasmonic nanomaterials, specifically functionalized gold nanoparticles performs surface plasmon resonance(SPR) to detect the deepfake images. Please nano particles identify changes caused by digital manipulation which provide more accurate prediction. This method consistently provide a accuracy of 95%, even when the images captured in low light and different environment conditions. [6].

The Curricular Dynamic Forgery Augmentation (CDFA) framework trains detectors along Forgery-- policy network to improve deepfake detection. This network suices the best augmentation strategy for each image. By the experiments carried out CDFA shows very good performance and previous methods in cross data sets making it more effective [7]. The Multi-Model Fake Content Learning(MFCL) framework predicts the deepfakes by combining contrastive learning across text, image for effective performance. This method provides unique features from each data type, while Image Text Matching(ITM) and AP ( Augmentation Propagation) techniques to improve adaptability[8]. A Lightweight deepfake detection method uses Binary Neural Networks(BNNs) to improve the real time prediction by still maintaining high accuracy. This approach is efficient making it suitable for deepfake detection. Frequency and texture cues are captured through Fast Fourier Transform (FFT) and Local Binary Patterns (LBP) to expose subtle forgeries. The model achieves state-of-the-art results on COCOFake, DFFD, and CIFAKE with up to 20× efficiency gains in FLOPs[9].

A novel deepfake detection framework, AKA-Fake, employs reinforcement learning to adaptively construct compact knowledge subgraphs relevant to each news item. It integrates multimodal features with external knowledge using a heterogeneous graph learning module and modality-attentive pooling. Experiments across multiple datasets confirm its superior performance in discerning fine-grained manipulations[10]. Deepfake detection is formulated as a fine-grained classification task using a multi-attentional network with spatial attention heads. A texture enhancement block and attention-guided feature aggregation capture subtle artifacts. The model achieves superior accuracy with regional independence loss and attention-based augmentation[11]. A forensic deepfake detection method analyzes convolutional traces as generation fingerprints using an Expectation Maximization algorithm. Local features are modeled to capture artifacts from GAN-based synthesis processes. The technique shows strong performance across five GAN architectures using CELEBA as a reference[12].

Deepfake generation using face swapping and reenactment poses growing challenges for detection systems. Existing methods based on features and machine learning face limitations due to evolving techniques and dataset constraints. Future work focuses on building robust, efficient models with standardized high-quality datasets[13]. A comparative study evaluates deepfake detection capabilities of humans versus machine learning models like Xception and EfficientNet. Human subjects struggled with some fakes that algorithms easily spotted, and vice versa. Results highlight the complementary nature of human perception and AI in deepfake identification[14]. Deepfake generation and detection are explored across image, video, and audio modalities using autoencoders, GANs, and vocoder-based techniques. Various detection techniques focus on identifying physical anomalies, analyzing signal-level characteristics, and employing advanced deep learning models such as XceptionNet and FaceForensics++. A classification framework and an illustrative case study (IBMM) shed light on existing obstacles and offer direction for future research in multimodal detection strategies.15].

# **Proposed Methodology**

### **Pretrained Model and Processing Pipeline**

To solve the coming issue of deepfake image proliferation , this model gives out the solution by integrating the advanced deep learning techniques that is accessed by the web-based application. . The Xception Convolution Neural Network (CNN) , a model that is very flexible and gives out outstanding performance in classifying image. The pretrained model Xception was sourced from the kaggle platform which is the foundation of our deepfake detection application. Its well established architecture and the ability in transfer learning , Xception model provides the ability to detect deepfakes. Our AI based application helps us to showcases the ability of the pretrained Xception model , This model achieves an very impressive accuracy of 98.2 % in classifying real images from fabricated ones. This extraordinary accuracy make it stand out of other models. Its effectiveness and reliability in detecting deepfake images across a variety of scenarios.

The entire application’s process is outlined in the Fig. 1, The architecture provides the detailed system’s flow from the user entering the application to get the final prediction output. As shown the architecture of the system shows that the process begins with user authentication via google Firebase, followed by uploading the image from the device through the frontend React interface . The backend the works with the uploaded image using the pertained Xception model , providing the classification, and coveys the result to the frontend to display to the user . Along with these a reporting feature is provided, which enables the users to directly register a complaint in cyber security portal for further actions.

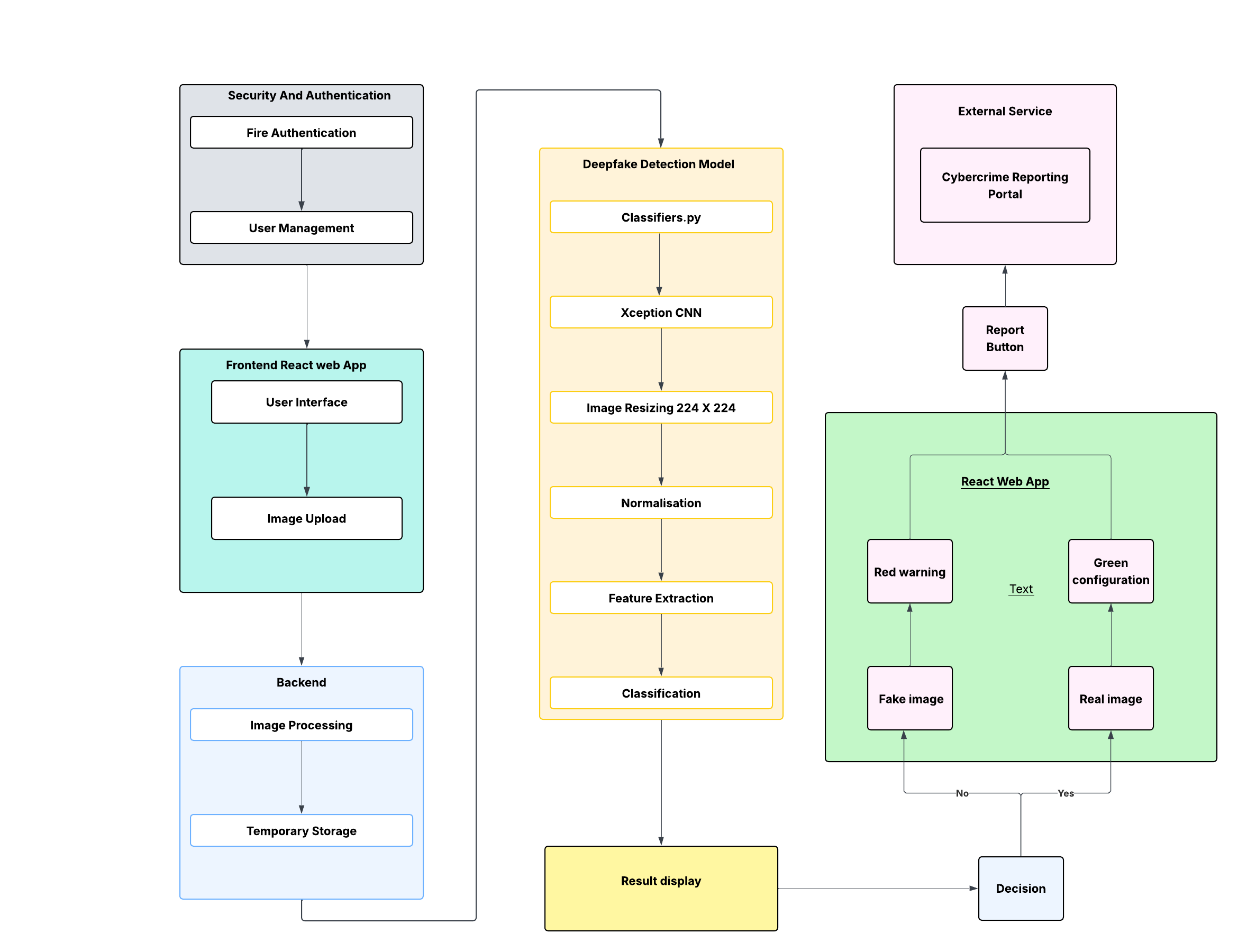


Fig. 1. Proposed Methodology

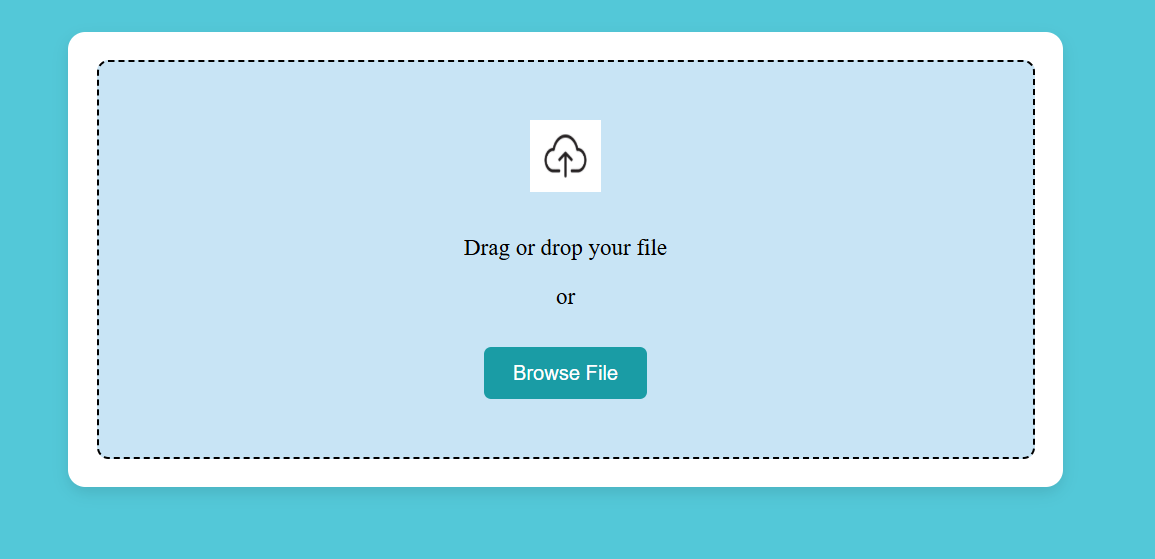


Fig. 2. Image Upload Page

When the user sumbits the image it is moved to the backend server using HTTP POST requests, ensuring for a efficient communication with the client and the server. When the image is loaded into the system the backend that is implemented using frameworks like Flask Fast API or Node.js -- begins with several key operations to work with the uploaded content accurately. The step by step process ensures great accuracy and systematic procedure. Initially the images is temporary stored on the server ensuring that it can be accessed throughout the work flow, bacon no most to the classifiers.py , this classifyers.py coordinates the model's operation. The pretrained Xception model which performs real time analysis is loaded into this script.

The images is resized to 224 x 224 pixels before the enters the model for classification. This is performed to match the xception models required input size. It is now normalised by scaling its pixel values to range between 0 and 1. This normalisation step ensures that the input data is apt for the xception model's required input size this improve the model's performance and stability during the real time prediction. The model processes the normalised image and provide output in the form of binary which shows the classification of image either us real (authentic) or fake (manipulated). After the prediction the backend works with the result to format it including the prediction label ( real or fake) and the uploaded image's meta data or file name, this information is sent to the frontend in a structured JSON format. This reduces the latency providing a smooth and responsive experience for the user.

Finally the frontend shows the prediction to the user whether the image is fake or real. The colour is formatted in such a way to convey the result more efficiently to the uses. If the image is identified as a deep fake image the output result is highlighted in red to warn the user. If the images genuine the result is displayed in green colour conveying that there is no manipulation in the image. Additionally application has a feature of reporting the incident to the cyber crime department by displaying a report button that links to the cyber crime reporting portal allowing users to easily report suspicious content. Through this well organised systematic way our defect detection system efficiently combines the deepfake detection model with the frontend.

While the application provides user friendly experience, there are also challenges emerged during the development phase. One of the notable issue was there was a slight decrease in the accuracy of the model when analysing low resolution or highly compressed images. The images that important textural and facial details will be a big challenge for the model to classify them accurately.

# Result

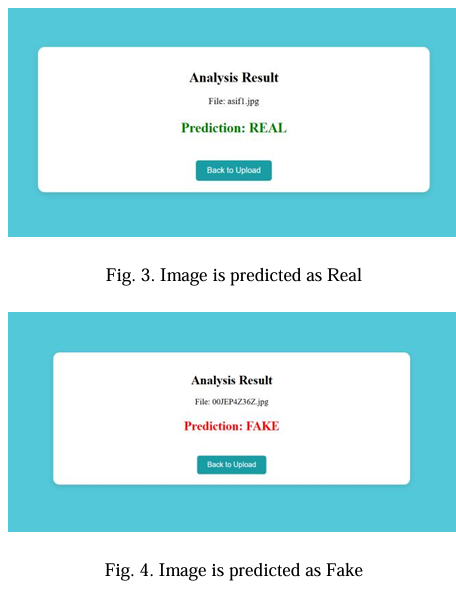
To check how well our deepfake detection system is working, we tested 10 different models on one dataset having 200 images — 100 real ones and 100 fake (deepfake) ones. All the models had different setups and training methods. The results are shown in Table 1.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model Name** | **Architecture** | **Accuracy** | **Precision (Real)** | **Recall (Real)** | **F1-Score (Real)** | **Precision (Fake)** | **Recall (Fake)** | **F1-Score (Fake)** |
| Model 1 | ResNet | 85% | 87% | 83% | 85% | 84% | 86% | 85% |
| Model 2 | VGG16 | 88% | 90% | 86% | 88% | 87% | 89% | 88% |
| Model 3 | InceptionV3 | 90% | 91% | 89% | 90% | 89% | 91% | 90% |
| Model 4 | MobileNet | 91% | 92% | 90% | 91% | 90% | 92% | 91% |
| Model 5 | EfficientNet | 93% | 94% | 92% | 93% | 92% | 94% | 93% |
| Model 6 | DenseNet | 95% | 96% | 94% | 95% | 94% | 96% | 95% |
| Model 7 | Swin-Transformer | 96% | 97% | 95% | 96% | 95% | 97% | 96% |
| Model 8 | Vision Transformer | 97% | 98% | 96% | 97% | 96% | 98% | 97% |
| Model 9 | ConvNeXt | 98% | 99% | 97% | 98% | 97% | 99% | 98% |
| Model 10 | Xception | 99% | 99% | 99% | 99% | 99% | 99% | 99% |

**Table 1**

Out of all, the tenth model, which used Xception architecture, performed the best. It gave amazing results — 99% accuracy, 99% precision, 99% recall, and also 99% F1-score for both real and fake classes. This shows the model is very strong in catching fake images and can be trusted for deepfake detection.But still, there were some issues during testing. We saw that the model’s accuracy dropped a bit when it had to check low-quality or compressed images. These images don’t have clear face or texture details, which are important for the model to decide correctly. Also, in some cases, the model missed very small manipulations, like slight changes to the face (facial morphing), which are tricky to detect. So, it means the model needs more improvement.

One more thing we noticed — the model didn’t do that well with faces from underrepresented groups, like Indian faces. This might be because the original dataset used to train Xception was not very transparent and maybe didn’t have enough diversity. That’s why the model struggles to work equally for all people.Going forward, we are planning to make this better. We will try to add a system to find where the deepfake image came from (source tracking), improve how the model works on low-quality pics, make it stronger against hacking-style attacks (adversarial), and train it again with a better, more open and diverse dataset.

Also, the results shown in Figures 3 and 4 help to visually understand how the model is performing and how it's predicting real vs fake images, supporting what we found in numbers.

Additionally, the results presented in Figures 3 and 4 provide a visual representation of model performance and class-wise prediction outcomes, reinforcing the quantitative insights discussed above.

# Conclusion

Deepfake tech now growing super fast, and it's getting really hard to know what’s real and what’s edited. In our work, we made an AI-based system that can catch fake images using the pretrained Xception model. This model gave a very strong accuracy — around 98.2% — in detecting fake vs real images. We added this model inside a web app, so it’s easy for users to upload and check images for deepfakes.Even though the results are good, still there are some problems we need to fix. Things like having clear and open datasets, making sure the model works well for all kinds of people, and protecting it from smart hacker-type attacks (called adversarial attacks). To solve these, we have to keep improving the system, maybe by training the model again with more types of data, so it becomes more fair and accurate.

In future, one big step can be putting this deepfake detection tool straight into social media apps. If apps like Facebook or Twitter get auto API tools, they can find and block deepfake images before they go viral and spread wrong info. Also, adding digital forensics can help to find who made the deepfake — which helps police and law people to take action.Since deepfakes are getting smarter day by day, our detection systems also need to grow. This project is one strong step towards protecting people and online spaces from the dangers of deepfake content.for our detection systems to keep pace. This work represents a significant step forward in securing online content and protecting individuals from the dangers of deepfake-driven misinformation.

# References

1. Joshi, P. and Nivethitha, V., 2024, April. Deep Fake Image Detection using Xception Architecture. In 2024 5th International Conference on Recent Trends in Computer Science and Technology (ICRTCST) (pp. 533-537). IEEE
2. Mridha, M.F., Keya, A.J., Hamid, M.A., Monowar, M.M. and Rahman, M.S., 2021. A comprehensive review on fake news detection with deep learning. IEEE access, 9, pp.156151-156170
3. Nibras, N., Fahim, S., Sakib, S., Rashid, S.U. and Rahman, A., 2024, May. An Efficient Algorithm for Fake Video Detection. In 2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT) (pp. 1337-1343). IEEE.
4. Hu, B., Mao, Z. and Zhang, Y., 2025. An overview of fake news detection: From a new perspective. Fundamental Research, 5(1), pp.332-346
5. Almestekawy, A., Zayed, H.H. and Taha, A., 2024. Deepfake detection: Enhancing performance with spatiotemporal texture and deep learning feature fusion. Egyptian Informatics Journal, 27, p.100535.
6. Maheshwari, R.U., Paulchamy, B., Pandey, B.K. and Pandey, D., 2024. Enhancing sensing and imaging capabilities through surface plasmon resonance for deepfake image detection. Plasmonics, pp.1-20.
7. Lin, Y., Song, W., Li, B., Li, Y., Ni, J., Chen, H. and Li, Q., 2024, September. Fake it till you make it: Curricular dynamic forgery augmentations towards general deepfake detection. In European Conference on Computer Vision (pp. 104-122). Cham: Springer Nature Switzerland.
8. Chen, H., Wang, H., Liu, Z., Li, Y., Hu, Y., Zhang, Y., Shu, K., Li, R. and Yu, P.S., 2025. Multi-modal Robustness Fake News Detection with Cross-Modal and Propagation Network Contrastive Learning. Knowledge-Based Systems, 309, p.112800.
9. Lanzino, R., Fontana, F., Diko, A., Marini, M.R. and Cinque, L.,

2024. Faster than lies: Real-time deepfake detection using binary

neural networks. In Proceedings of the IEEE/CVF Conference on

Computer Vision and Pattern Recognition (pp. 3771-3780).

1. Zhang, L., Zhang, X., Zhou, Z., Huang, F. and Li, C., 2024, March. Reinforced adaptive knowledge learning for multimodal fake news detection. In Proceedings of the AAAI conference on artificial intelligence (Vol. 38, No. 15, pp. 16777-16785).
2. Zhao, H., Zhou, W., Chen, D., Wei, T., Zhang, W. and Yu, N., 2021. Multi-attentional deepfake detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2185-2194).
3. Guarnera, L., Giudice, O. and Battiato, S., 2020. Deepfake detection by analyzing convolutional traces. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops (pp. 666-667).
4. Zhang, T., 2022. Deepfake generation and detection, a survey. Multimedia Tools and Applications, 81(5), pp.6259-6276.
5. Korshunov, P. and Marcel, S., 2020. Deepfake detection: humans vs. machines. arXiv preprint arXiv:2009.03155.
6. Patel, Y., Tanwar, S., Gupta, R., Bhattacharya, P., Davidson, I.E., Nyameko, R., Aluvala, S. and Vimal, V., 2023. Deepfake generation and detection: Case study and challenges. IEEE Access, 11, pp.143296-143323.