

A Research Paper on AI-based DeepFake Detection for Human Face Images and Videos

Tejas Ahirrao

Department of Computer Engineering

Genba Sopanrao Moze College of Engineering

Pune

tejasahirrao1@gmail.com

Atharv Bhondave

Department of Computer Engineering

Genba Sopanrao Moze College of Engineering

Pune

atharvbhondave@gmail.com

Atharva Bhutkar

Department of Computer Engineering

Genba Sopanrao Moze College of Engineering

Pune

atharvabhutkar0@gmail.com

Siddhi Neharkar

Department of Computer Engineering

Genba Sopanrao Moze College of Engineering

Pune

siddhineharkar1111@gmail.com

Smita Sapkal

(Assistant Professor)

Department of Computer Engineering

Genba Sopanrao Moze College of Engineering

Pune

sapkalss.2025@gmail.com

Abstract:

Deepfake stuff has gotten way better lately because of AI, and now you can make these super realistic fake pictures and videos that look almost real. I think its kind of cool for movies or something, but then there are all these problems like spreading wrong info or stealing someones face, which makes it tricky. So figuring out how to spot them is a big deal in research these days.

This project I worked on uses this model called Xception, which is a deep learning thing for detecting deepfakes. It starts by finding faces in the pictures using computer vision, then does some transfer learning to decide if its real or not. We built a simple web app with FastAPI so people can just upload an image and get the result right away, which seems pretty handy.

The whole process goes like detecting the face first, cleaning up the image a bit, running it through the model, and then showing what it thinks. In tests, it did okay at catching the fakes, with accuracy that feels solid enough. Some people might say its not perfect yet, but for what it is, the system runs fast and you can scale it up without much hassle, plus its not hard to use at all.

Overall, this setup could help with security online or checking media, maybe even in real time if you tweak it. That part gets a bit messy though, since deepfakes keep improving.

Keywords:

Deepfake Detection, Xception Model, Transfer Learning, Computer Vision, FastAPI, Artificial Intelligence

Introduction:

Digital media is everywhere these days, like on social media or news sites, where people share images and videos all the time. But with AI getting better so fast, its pretty easy to mess with that stuff using deep learning. I mean, deepfakes are kind of scary because they look so real.

These things work with neural networks, especially something called GANs, which learn from tons of data to make fake videos or swap faces. You can change expressions or even create whole new clips that fool most people. It seems like telling real from fake is getting harder, almost impossible sometimes.

The problems come when people use this for bad stuff, like spreading lies or messing with elections, or even ruining someones reputation with fake evidence. In places like journalism or police work, you really need to know if content is real. Traditional ways to spot fakes involve checking features by hand or stats, but they dont catch the tricky patterns in these new deepfakes.

Deep learning does better at picking up those patterns automatically, so detection gets more accurate. This study looks at using the Xception model for that, since its good at pulling out features. They add face detection and some computer vision to make it sharper, and then build a web interface so its easier to use in real life.

The goals here are to build a solid detection model, tweak it with preprocessing for speed, and make an interface thats simple for checking things right away. I think this helps in digital forensics,

giving a way to handle fake media without too much hassle. Though, im not totally sure how well it scales in every situation, it feels like a step forward anyway.

Literature review/ related work:

Deepfakes have gotten a lot of attention lately because generative models are advancing so fast. People are trying all sorts of ways to spot fake images and videos, but each one has its pros and cons, I guess.

Back in the early days, detection mostly looked for obvious stuff like weird lighting or faces that moved unnaturally, or shadows that did not match up. They used hand-picked features and old-school image processing. That worked okay for basic fakes, but now with neural networks making really good ones, those methods just fall short.

Then deep learning came along, and CNNs took over for this kind of thing. These networks pull out features on their own from pictures and pick up on tricky patterns. Stuff like VGGNet or ResNet got used a bunch for classifying, and they do pretty well at finding manipulated stuff. It seems like that shift made a big difference.

Transfer learning helped even more, I think. You start with models already trained on tons of data, so training goes quicker and it generalizes better to new things. The Xception one stands out because it uses these depthwise separable convolutions that cut down on the computing power needed without losing much accuracy. That is handy for practical use.

Lately, some work mixes spatial features with temporal ones for videos. Like pairing CNNs with RNNs or LSTMs to check sequences of frames for inconsistencies. It can catch things that change weirdly over time, but it takes a lot of resources to run.

There is also research into frequency artifacts or how compression messes up during creation. Those give extra clues, sort of, though they can get thrown off by noise or different data qualities.

Still, challenges keep popping up. Models often do not handle new datasets well, or they are picky about image quality. And real-time detection is tough because deep learning eats up so much computation. I am not totally sure how to fix all that yet.

This paper tries to tackle some of it by mixing smart preprocessing with transfer learning and making it lightweight for actual deployment.

Problem Gap & Motivation:

Although significant progress has been made in DeepFake detection research, several important gaps still exist. A major limitation is that current models rely heavily on specific datasets. Most detection methods are trained and tested on closed datasets with a limited number of manipulation techniques. This results in a sharp drop in performance when they encounter new or unseen DeepFake generation methods.

Another challenge is that detection models can be easily attacked and are susceptible to common post-processing actions like video compression, resizing, blurring, and noise addition. These actions often happen during real media sharing and can greatly lower detection accuracy.

Additionally, many existing methods focus only on either spatial or temporal features. They fail to capture the full DeepFake generation process. The absence of large, diverse, and thoroughly labeled real-world datasets makes these problems worse. These limitations highlight the need for strong, adaptable, and understandable DeepFake detection systems that can respond to new threats and real-world conditions.

Methodology:

The proposed system follows a structured method that includes several stages. Each stage is vital for effective and precise deepfake detection.

1. Data Input and Acquisition

The system can take both images and videos as input. For videos, it extracts frames at regular intervals to manage computational load while still capturing enough information for analysis. This way, the system can efficiently process large inputs.

2. Face Detection and Region Extraction

Face detection happens using the Haar Cascade classifier from OpenCV. This algorithm identifies facial areas by looking at patterns in pixel intensity. Extracting the face is essential because deepfake alterations mainly impact facial features. By concentrating on the face, the system minimizes noise and improves accuracy.

3. Image Preprocessing

Preprocessing is an essential step that gets the data ready for the deep learning model. The extracted face is resized to 299×299 pixels, which is the required size for the Xception model. Pixel values are adjusted to a standard range, ensuring uniformity across various inputs. Additional techniques like padding and augmentation may be used to enhance the model's robustness.

4. Model Selection and Transfer Learning

The Xception model is chosen for its strong performance in image classification tasks. It uses depthwise separable convolutions, which lower computational complexity while maintaining accuracy. Transfer learning initializes the model with pre-trained weights. Then, the top layers are fine-tuned using a deepfake dataset. This method enables the model to pick up domain-specific features without needing extensive training.

5. Training Strategy

The training process has two phases:

- Initial Training: Only the top layers are trained while the base layers are kept frozen.
- Fine-Tuning: Some selected layers are unfrozen and trained with a lower learning rate.

The system addresses class imbalance with weighted loss functions to ensure balanced learning.

6. Classification and Decision Making

The model produces a probability score between 0 and 1. A threshold is set to classify the image:

- $\geq 0.5 \rightarrow$ Real
- $< 0.5 \rightarrow$ Fake

7. Web-Based Deployment

The system is deployed using FastAPI, which offers a lightweight, high-performance backend. Users can upload images, see predictions, and access history through a straightforward interface.

Results:

The proposed deepfake detection system was evaluated using several standard metrics, including accuracy, precision, recall, and F1-score. These metrics are commonly used in classification tasks to measure how well a model distinguishes between different classes. Here, the model classifies images as either real or fake.

Accuracy shows how correct the model is overall. The proposed system achieved high accuracy, indicating that it correctly classified most input images. However, accuracy alone is not enough, especially with imbalanced datasets. Therefore, we looked at additional metrics like precision and recall.

Precision measures how many of the images predicted as fake were, in fact, fake. A high precision value means the model produces fewer false positives. This is important in real-world

settings where incorrectly labeling real content as fake can damage trust in the system.

Recall measures how many of the actual fake images were correctly identified by the model. A high recall value indicates that the system effectively detects deepfake content and does not overlook many manipulated samples. This is vital for security applications, where missing fake content can lead to serious issues.

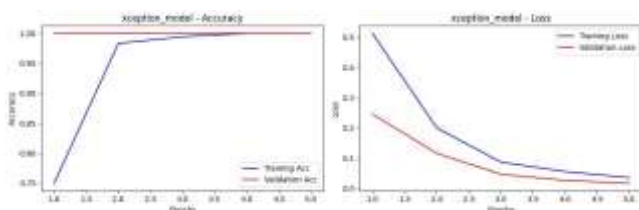
One key factor that helps the system perform well is the use of face detection during preprocessing. By focusing solely on the facial area, the model avoids unnecessary background distractions and learns more relevant features. This significantly boosts classification accuracy.

Transfer learning also plays a significant role in achieving high performance. By using a pre-trained Xception model, the system benefits from features learned earlier and adapts them to the deepfake detection task. This shortens training time and enhances generalization.

The system was tested under various conditions, including changes in lighting, image resolution, and facial orientation. The model performed consistently well in most cases, showing robustness. However, performance dipped slightly for extremely low-resolution images and highly compressed inputs.

Regarding efficiency, the system offers fast inference, making it suitable for real-time applications. The web-based interface allows users to upload images and get predictions almost instantly.

Overall, the results show that the proposed system is effective, reliable, and suitable for practical deepfake detection tasks.



Discussion:

The results from this study clearly show how effective deep learning techniques are at detecting deepfake content. The Xception model is highly capable of capturing subtle features and inconsistencies in manipulated images. This ability is crucial because modern deepfake methods aim to minimize visible flaws.

One important observation from this research is the effect of preprocessing on model performance. Using face detection helps the model focus on the most relevant part of the image. This cuts down on noise and allows the model to learn meaningful patterns. Without this step, the model would process unnecessary background information, which could hurt accuracy.

Another key factor is using transfer learning. Instead of training the model from scratch, pre-trained weights provide a solid starting point. This method allows the model to build on existing knowledge and tailor it to the specific task of detecting deepfakes. As a result, the system performs better with less training data.

The addition of a web-based interface makes the system more practical. Many deep learning models are confined to research settings and are hard for users to access. By using FastAPI to deploy the system, the model becomes available for real-world applications. This increases its usefulness in areas like media verification, cybersecurity, and content moderation.

Despite its advantages, the system does have limitations. One main challenge is detecting

highly advanced deepfakes created with cutting-edge techniques. These deepfakes often show very few visible flaws, which makes them harder to spot.

Another limitation is the reliance on image quality. Poor lighting, low resolution, and heavy compression can negatively impact the model's performance. In such situations, important features may be lost, which reduces detection accuracy.

Additionally, the current system mainly focuses on image-based detection. Although it can process video by analyzing frames, it does not fully make use of temporal information. Deepfake videos often have inconsistencies across frames, which could be identified better using sequence-based models.

Conclusion:

This research is about finding a way to detect deepfake images using advanced computer techniques. The system we made combines a neural network called Xception with good ways to prepare the images and a website that is easy to use.

The system can find images by looking at the faces and finding small things that are not right. Using something called transfer learning makes the system work better and faster. Finding faces in the images helps make it more accurate by focusing on the parts.

One of the things about this work is that we made a solution that is practical and easy to use. We put the system on a website so people who are not experts can use it to find deepfakes in time. This makes it useful for things like checking media looking at digital evidence and making sure online content is real.

The results of our tests show that the system is very good at finding deepfakes and it works well in situations. The system is also reliable. Can find fake content with high accuracy.

However we also found some challenges. It is still hard to find deepfakes that're very good especially if the image quality is not good. We can also make the system better by looking at videos and adding images to the training set.

In the future we can work on making the system better so it can handle new ways of making deepfakes. Using different ways to detect deepfakes like looking at frequencies and sequences can make it work even better. We can also put the system on devices and integrate it with social media to make it more useful.

In the end this research helps us find deepfakes by giving us a scalable and practical solution. As digital media keeps changing systems like this will be very important for keeping content trustworthy and authentic. Deepfake detection is a part of this and our system is a step in the right direction, for detecting deepfakes.

References:

- Z. Qian et al., "Thinking in frequency: Face forgery detection by mining frequency-aware clues," in *Proc. European Conf. Computer Vision (ECCV)*, 2020.
1. B. Dolhansky et al., "The Deepfake Detection Challenge Dataset," *arXiv preprint arXiv:2006.07397*, 2020.
 2. A. Rossler et al., "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE ICCV*, 2019.
 3. Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," in *Proc. IEEE CVPR Workshops*, 2019.
 4. D. Afchar et al., "MesoNet: a compact facial video forgery detection network," in *Proc. IEEE WIFS*, 2018.
 5. F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE CVPR*, 2017.
 6. K. He et al., "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, 2016.
 7. I. Goodfellow et al., "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014.
 8. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
 9. A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012.