International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 04 April 2024
Page No: 938 - 943
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0131

AI Simulated Media Detection for Social Media

Mr. S. Kingsley[1], Adithya S P[2], Badrinath Babu[3], Harish Vaithilingam A[4]
[1]Assistant professor, Easwari Engineering College, Gangiah Avenue, Ramapuram, Chennai, India.
[2,3,4]Student, Easwari Engineering College, Gangiah Avenue, Ramapuram Chennai, India.
Emails: *kingsley.s@eec.srmrmp.edu.in[1], adithya.s.p.5505@gmail.com[2], badrinathb1810@gmail.com[3], happyharish2002@gmail.com[4]*

Abstract
This paper discusses the use of artificial intelligence (AI) for detecting AI-simulated media on social media platforms. AI-generated content, like deepfake videos and synthetic images, poses a significant challenge to content moderation. The paper highlights the methods and technologies such as machine learning and deep learning models involved in identifying such content, emphasizing the importance of dataset quality. The paper offers a holistic view of the multifaceted approach required to address the challenge of AI-simulated media on social media platforms.

Keywords: *CNN - Convolutional Neural Network; ML - Machine Learning; BB - Bounding Box;*

1. Introduction
In the age of information and connectivity, social media platforms have become not only conduits for communication and expression but also battlegrounds where misinformation, disinformation, and manipulated media vie for attention. One particularly insidious form of manipulated content is AI-simulated media, encompassing deepfake videos, computer-generated images, and text generated by language models. As these AI-generated artifacts become more sophisticated, the challenge of discerning fact from fiction on social media has never been more pressing. This paper delves into the realm of AI-Simulated Media Detection for Social Media Platforms, exploring the growing need for robust, scalable solutions in the ongoing battle against the spread of deceptive and potentially harmful content. This paper will journey through the landscape of AI-simulated media detection, providing insights into the underlying technologies, the challenges posed by increasingly convincing simulated content on social media. [1-4]

1.1 Types of AI Simulated Media
AI simulated media refers to content that is created using artificial intelligence (AI) techniques to mimic or simulate human-generated content. This can encompass various forms of media, including text, voice, images, and video. The primary goal of AI Simulated media is to generate content that appears authentic and indistinguishable from content created by humans. However, it is important to note that AI simulated media can be used for both legitimate and potentially malicious purposes. [5-7]

1.1.1 AI-Generated Voice
AI algorithms can synthesize human-like voices, often referred to as text-to-speech (TTS) technology. These voices can read text aloud with natural intonation and pronunciation. AI-generated voices have legitimate applications in assistive technologies and entertainment, but they can also be misused to create fake voice recordings, impersonate individuals, or manipulate audio content. [8]

1.1.2 AI-Generated Images and Video
Deep learning models, like Generative Adversarial Networks (GANs), can produce images and videos that appear genuine but are entirely generated by AI. Deepfakes are a notable example of AI-generated video, where the likeness of individuals can be superimposed onto other people's bodies or faces. This technology has the potential to be used for entertainment and special effects, but it can also be used maliciously for creating misleading or harmful content. [9-12]

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 04 April 2024
Page No: 938 - 943
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0131

2. Detection Framework

2.1. Machine Learning and Deep Learning

Machine learning and deep learning techniques have emerged as powerful tools for detecting AI-simulated media. These methods involve training models on extensive datasets of both real and manipulated content. Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) are commonly used in deepfake detection. These models learn to recognize patterns, inconsistencies, or anomalies, making them effective at identifying manipulated media. [13]

2.2. Multimodal and Real-Time Detection

Multimodal detection combines various techniques, such as analyzing both visual and audio content simultaneously, enhancing overall accuracy. Real-time detection is essential for social media platforms, as it enables the identification of AI-simulated media as it is posted or shared. Achieving real-time detection requires not only sophisticated algorithms but also efficient infrastructure to handle the vast amount of content uploaded to these platforms. [14]

2.3. Web Extension Integration

Our project offers a distinct advantage over traditional detection methods by providing real-time AI-simulated media detection through the seamless integration of a web extension. Unlike conventional techniques that often involve post-analysis or off-platform processing, our solution operates within the user's web browser environment, ensuring that AI-generated content is identified and the user is warned as it appears on social media platforms. This real-time capability empowers users with protection against the deceptive AI-generated media, enhancing the overall safety and trustworthiness of their online experience. [15]

3. Method

3.1. CNN Algorithm

CNNs have been a powerful tool in computer vision tasks, and they can be adapted for identifying AI-generated content. Here's how CNNs can be applied to this problem. Figure 1 shows the Convolutional Neural Network.

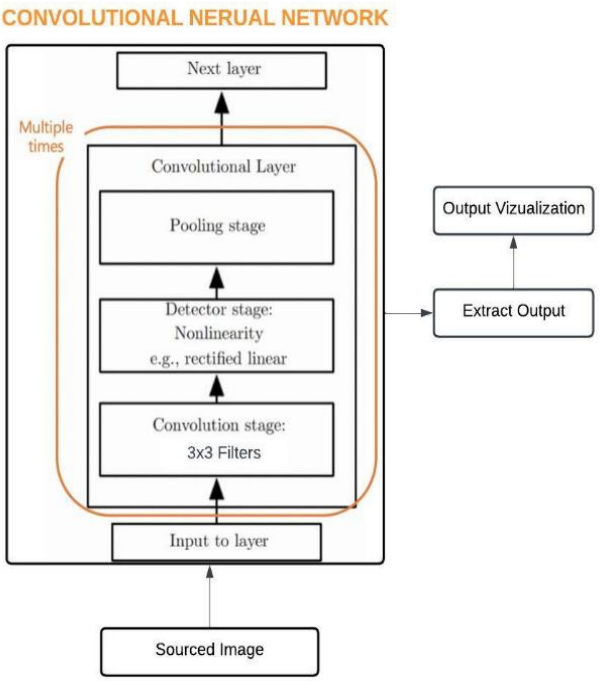

Figure 1 Convolutional Neural Network

3.1.1. AI-Simulated Image Detection

CNNs can be trained to recognize specific patterns, artifacts, or anomalies associated with AI-simulated images, such as deepfake facial features, inconsistencies in lighting and shadows, or telltale signs of manipulation. [16]

3.1.2. AI-Simulated Video Detection

CNNs can be used to analyze individual frames within a video to identify visual inconsistencies or artifacts that suggest AI-generated content.

3.1.3. Advantages

CNNs are highly effective at extracting features from visual data, making them suitable for spotting subtle visual differences or manipulations in images and videos. They can be trained on large datasets, which can help in recognizing various types of AI-simulated content. CNNs can operate in real-time or near real-time, making them suitable for use in web extensions for social media platforms. [17]

3.1.4. Limitations

CNNs might require significant computational resources for both training and inference, which can be a challenge for web extensions with limited resources. They are typically specialized for image

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 04 April 2024
Page No: 938 - 943
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0131

analysis and might not address other types of AI-simulated content, such as audio deepfakes. CNNs can perform well on known patterns and features but may struggle with detecting novel and previously unseen AI-simulated content.

4. Proposed Method

4.1. Content Monitoring

The web extension constantly monitors the media content being shared or consumed on social media platforms. It can hook into the user's web browser to access the content as it's displayed.

4.2. Data Acquisition

The extension collects and extracts relevant media content, such as images, videos, or audio, from the user's social media feed or the content being viewed.

4.3. Analysis and Processing

The extension uses image analysis algorithms, including Convolutional Neural Networks (CNNs) to scan for anomalies, visual artifacts, or inconsistencies that suggest AI-simulated content.

5. Meso-4

Our experimentation commenced with intricate architectures, progressively simplifying them until arriving at the subsequent model, which yields equivalent results but with greater efficiency. This network initiates with a sequence of four layers comprising consecutive convolutions and pooling, succeeded by a dense network featuring a single hidden layer. To enhance generalization, the convolutional layers employ ReLU activation functions to introduce non-linearities and incorporate Batch Normalization to regularize their output, thereby mitigating the vanishing gradient issue. Additionally, the fully-connected layers utilize Dropout for regularization, thereby enhancing robustness. An alternative configuration involves substituting the initial two convolutional layers of Meso4 with a variation of the inception module, as introduced by Szegedy et.al. The module's concept revolves around amalgamating the outputs of multiple convolutional layers with varying kernel shapes to broaden the function space optimized by the model. Instead of utilizing the original module's 5 × 5 convolutions, we suggest employing 3 × 3 dilated convolutions to mitigate high semantic redundancy. While the utilization of dilated convolutions within the inception module for multi-scale information processing has been proposed in prior works, we enhance this approach by incorporating 1×1 convolutions before dilated convolutions for dimension reduction. Figure 2 shows the Network Architecture of Meso-4 Layers and Parameters Are Displayed in the Boxes, Output Sizes next To the Arrows. Additionally, we introduce an additional 1×1 convolution in parallel, serving as a skip-connection between successive modules. Refer to Figure 3 for a comprehensive illustration of these modifications.

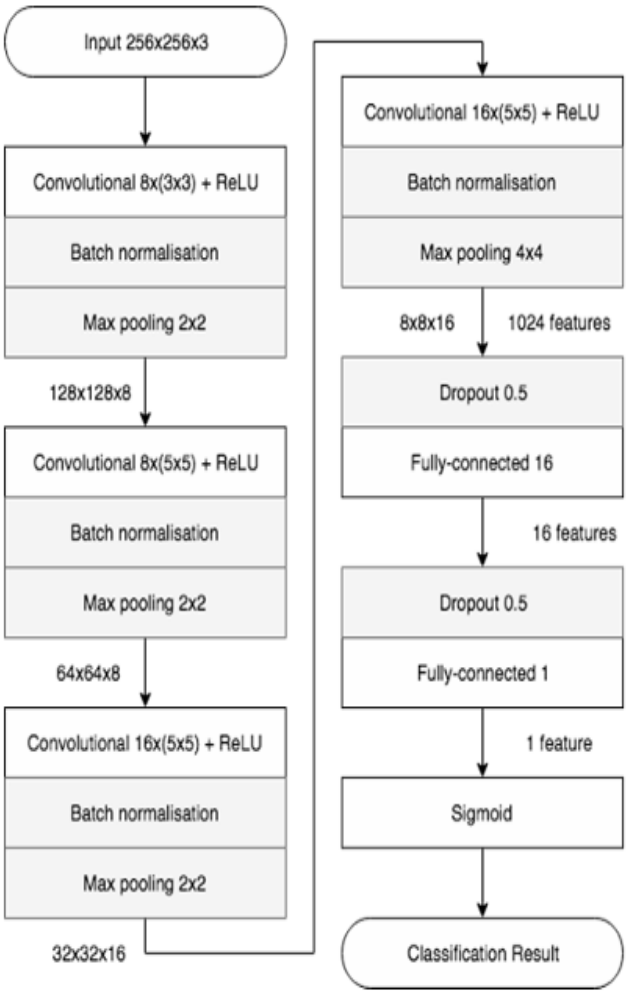

Figure 2 The Network Architecture of Meso-4 Layers and Parameters Are Displayed in the Boxes, Output Sizes Next to the Arrows

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 04 April 2024
Page No: 938 - 943
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0131

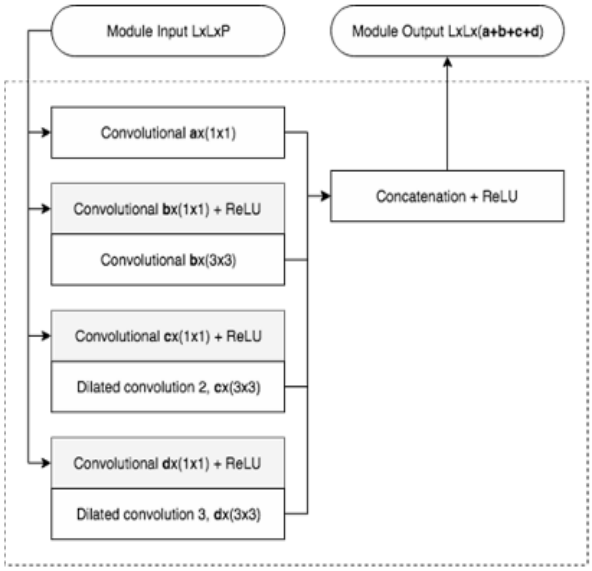

Figure 3 Architecture of the Inception Modules Used in Mesoinception-4 the Module is Parameterized using A, B, C, and D EN the Dilated Convolutions Are Computed without Stride

6. Haar Cascade

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle. Now, all possible sizes and locations of each kernel are used to calculate lots of features. (Just imagine how much computation it needs? Even a 24x24 window results over 160000 features). For each feature calculation, we need to find the sum of the pixels under white and black rectangles. To solve this, they introduced the integral image. However large your image, it reduces the calculations for a given pixel to an operation involving just four pixels. Nice, isn't it? It makes things super-fast. But among all these features we calculated, most of them are irrelevant. For example, consider the image below. The top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applied to cheeks or any other place is irrelevant. So how do we select the best features out of 160000+ features? It is achieved by Adaboost.

Integral Image: The first step in Haar feature computation is to convert the image into an integral image. The integral image representation allows for fast computation of the sum of pixel values within any rectangular area of the image. This is crucial for the efficient computation of Haar-like features. Haar Features: Haar features are simple rectangular filters that are applied to regions of an image to compute features. These features are calculated by subtracting the sum of pixel values in one region of the image from the sum of pixel values in another region. Haar features are defined by their location, size, and orientation.

Types of Haar Features: Haar features can be of different types, such as: Edge Features: These features capture changes in intensity along a line. Line Features: These features capture changes in intensity over a wider area. Center-Surround Features: These features compare the average intensity in a central region to the average intensity in a surrounding region.

Adaboost: Adaboost (Adaptive Boosting) is a machine learning algorithm used to select a small number of important features from a large pool of potential features. It works by iteratively training a weak classifier on the training data, with each subsequent classifier focusing more on the examples that the previous classifiers misclassified. The final classifier is a weighted combination of these weak classifiers.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 04 April 2024
Page No: 938 - 943
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0131

Cascade Classifier: The Haar cascade classifier is formed by combining multiple weak classifiers into a cascade. Each weak classifier focuses on a specific subset of features and is trained to determine whether a particular region of the image contains the object of interest (e.g., a face). The cascade structure allows for fast rejection of regions of the image that are unlikely to contain the object, leading to significant speed improvements during detection.

Training: Training a Haar cascade classifier involves two main steps:

Positive Sample Collection: Collect a large number of images containing the object of interest (e.g., faces) and extract positive examples from these images.

Negative Sample Collection: Collect images that do not contain the object of interest and extract negative examples from these images. Training Process: Use Adaboost to select a subset of the most discriminative Haar-like features and train the cascade classifier using these features and the positive and negative samples.

Detection: Once the Haar cascade classifier is trained, it can be used for object detection in images or videos. The classifier slides over the image at multiple scales and locations, applying the cascade of weak classifiers to each region of the image. If a region passes all stages of the cascade, it is considered to contain the object of interest.

7. Web Extension

The envisioned web extension aims to combat the proliferation of manipulated and synthetic media on social media platforms by leveraging artificial intelligence (AI) technology. Upon installation, the extension will seamlessly integrate into users' web browsers, operating discreetly in the background to provide real-time detection and identification of AI-generated content, including deepfakes, altered images, and fabricated videos. The functionality of the extension relies on a sophisticated AI detection system, meticulously crafted to discern between authentic and manipulated media forms. As users browse social media platforms, the extension continuously scans the content displayed on their feeds, scrutinizing images and videos for telltale signs of AI manipulation. Behind the scenes, the extension employs advanced algorithms, machine learning models, and deep neural networks to analyze media content for anomalies indicative of AI-generated alterations. These algorithms are trained on vast datasets comprising both genuine and manipulated media, enabling the system to recognize subtle deviations from authentic content with high accuracy. Upon detecting potentially manipulated media, the extension alerts users through intuitive visual indicators or informative pop-up notifications, prompting them to exercise caution when engaging with the flagged content. Additionally, users may have the option to access detailed analysis reports, providing insights into the specific techniques used in the manipulation and the likelihood of authenticity. To ensure seamless integration into users' browsing experiences, the extension operates with minimal intrusion, preserving the fluidity of their interactions on social media platforms. It functions as a silent guardian, working tirelessly in the background to safeguard users against the dissemination of deceptive content. In addressing the challenges inherent in combating AI-simulated media, the extension remains adaptable and responsive to the evolving landscape of digital deception. Regular updates and enhancements will be rolled out to keep pace with advancements in AI technology and emerging manipulation techniques, maintaining the efficacy of the detection system over time. Overall, the web extension endeavors to contribute to the ongoing discourse on maintaining the integrity and trustworthiness of information shared on social media platforms. By empowering users with tools to identify and discern manipulated content, it fosters a more informed and vigilant online community, resilient against the spread of misinformation and deception.

Conclusion

Detecting AI-generated media on social platforms is a crucial issue. It requires a diverse approach, including machine learning, image and audio analysis, semantic analysis, content fingerprinting, blockchain verification, pattern recognition, and user behavior analysis. Ensembles and deep learning

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 04 April 2024
Page No: 938 - 943
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0131

models enhance detection accuracy, but the choice depends on requirements and content. Combining techniques improves accuracy. Consider training data, resources, and ethics. Collaboration is vital for effective solutions. Transparency, responsible AI use, and user education are key to combating AI-generated content.

References

[1]. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial networks,Commun. ACM, vol. 63, no. 11, pp. 139144, 2020.

[2]. D. P. Kingma and M. Welling, Auto-encoding variational Bayes, 2013, arXiv: 1312.6114.

[3]. D.J.Rezende,S.Mohamed,andD.Wierstra, Stochastic Backpropagation and approximate inference in deep generative models, in Proc. Int. Conf. Mach. Learn., 2014, pp. 12781286.

[4]. R. Wu, G. Zhang, S. Lu, and T. Chen, Cascade EF-GAN: Progressive facial expression editing with local focuses, in Proc. IEEE/CVF Conf.Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 50215030.

[5]. Y. Shen, J. Gu, X. Tang, and B. Zhou, Interpreting the latent space of GANs for semantic face editing, in Proc. IEEE/CVF Conf. Comput. Vis.Pattern Recognit. (CVPR), Jun. 2020, pp. 92439252.

[6]. C.-H. Lee, Z. Liu, L. Wu, and P. Luo, MaskGAN: Towards diverse and interactive facial image manipulation, in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 55495558.

[7]. F. Matern, C. Riess, and M. Stamminger, Exploiting visual artifacts to expose deepfakes and face manipulations, in Proc. IEEE Winter Appl. Comput. Vis. Workshops (WACVW), Jan. 2019, pp. 8392.

[8]. L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo, Face X-ray for more general face forgery detection, in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 50015010.

[9]. A.Rossler, D.Cozzolino, L.Verdoliva, C. Riess, J. Thies, and M. Niessner, FaceForensics: Learning to detect manipulated facial images, in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 1 11.

[10]. X. Wu, Z. Xie, Y. GAO, and Y. Xiao, SSTNet: Detecting manipu lated faces through spatial, steganalysis and temporal features, in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2020, pp. 29522956.

[11]. Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, Thinking in frequency: Face forgery detection by mining frequency-aware clues, in Proc. Eur.Conf. Comput. Vis. Cham, Switzerland: Springer, 2020, pp. 86103.

[12]. B. Dolhansky, J. Bitton, B. P aum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, The DeepFake detection challenge (DFDC) dataset, 2020, arXiv:2006.07397.

[13]. Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, Celeb-DF: A large-scale challenging dataset for DeepFake forensics, in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 32073216.

[14]. D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales, Learning to generalize: Meta-learning for domain generalization, in Proc. AAAI Conf. Artif. Intell. 2018, vol. 32, no. 1, pp. 18.

[15]. K. Hsu, S. Levine, and C. Finn, Unsupervised learning via Meta learning, 2018, arXiv: 1810.02334.

[16]. Z. Li, F. Zhou, F. Chen, and H. Li, Meta-SGD: Learning to learn quickly for few-shot learning, 2017, arXiv: 1707.09835.

[17]. R. Natsume, T. Yatagawa, and S. Morishima, RSGAN: Face swapping and editing using face and hair representation in latent spaces, 2018, arXiv: 1804.03447.