Deep Learning Technique Application for The Detection of Synthetic Images

Okpala Calista Uchenna
Department of Computer Science, Federal Polytechnic Oko, Anambra State
okpalacalista23@gmail.com

DOI: 10.56201/ijcsmt.v11.no1.2025.pg12.20

Abstract

This study presents a novel approach for detecting synthetic images using a Convolutional Neural Network (CNN). The proposed approach makes use of a two-step procedure: first, data collection and preprocessing; next, model training and assessment. While pre-trained diffusion and Generative Adversarial Network (GAN) models were used to create synthetic images, real images were used from publicly available datasets such as FFHQ, AFHQ, and LSUN. To differentiate between genuine and artificial photos, a CNN model with a complex architecture was created. It consists of fully connected layers for classification and convolutional layers for feature extraction. Using a 10-fold cross-validation method, the system's average accuracy, precision, recall, and F1-score were 96.7%, 0.96, and 0.97, respectively. The results obtained show how well the model detects synthetic images with high recall and precision, underscoring its potential for practical uses in content authentication, digital forensics, and AI-generated image recognition. The study emphasises how crucial it is to use deep learning methods to tackle the escalating difficulties in synthetic image identification.

Keywords: Synthetic Image Detection; Convolutional Neural Network; Deep Learning; Data Preprocessing; Image Classification; Generative Adversarial Networks (GANs)

1. INTRODUCTION

Recent advances in deep learning, particularly in Generative Adversarial Networks (GAN) (Goodfellow et al., 2014), have made it possible to create artificial photographs that are so good that they frequently pass for real ones (Masood et al., 2022). Because of the potential for abuse, this development presents serious ethical concerns even as it creates intriguing prospects across a range of businesses. These misuses might have significant social repercussions, such as spreading misleading information, committing fraud, stealing identities, and producing offensive or damaging content (Mirsky and Lee, 2021). Establishing reliable methods to distinguish between actual and artificial intelligence-generated information is crucial as deep learning advances at a never-before-seen rate and AI-generated visuals becoming more realistic by the day. This is necessary to mitigate the negative consequences and preserve the reliability and accuracy of information found online.

There are now two primary methods for identifying phoney photographs, aside from eye inspection, which is an inaccurate technique. Among these are deep learning and image processing methods for manually created feature extraction (Masood et al., 2022). Early techniques for identifying tampered photos sought to pinpoint a particular tampering approach, such copy-move or splicing (Thakur and Rohilla, 2020). In order to generate or edit things inside a image, these tampering techniques usually include changing specific portions of the image. The majority of early fake-detection techniques relied on frequency-domain feature extraction. For instance, the method suggested in (Fridrich et al., 2003) splits the image into overlapping blocks and uses discrete cosine transformation to match the characteristics taken from these blocks in order to identify copy-move forgeries.

Similarly, the discrete wavelet transformation is used to acquire low-frequency components in the feature extraction from the frequency domain approach provided in Li et al. (2007). To get the feature vectors, these components are simultaneously subjected to singular value decomposition. Nevertheless, this approach is laborious and susceptible to blurred, scaled, or twisted image objects. The same author suggested using the Fourier-Mellan transformation in (Bayram et al., 2009) to increase the effectiveness of the earlier approach. The Bloom filter speeds up the whole detection process while transformation reduces sensitivity to geometric operations. However, when it comes to image splicing, the aforementioned techniques fall short since it entails combining segments from many sources, each of which has unique textures and characteristics.

The colour filter array pattern is extracted from the image using the technique suggested in Ferrara et al. (2012). To distinguish between real and phoney areas, local irregularities within these patterns are subsequently statistically analysed. In order to detect the phoney images, the classification model is utilised in He et al. (2012) to detect abnormalities generated by splicing in images. Since splicing often alters the original image's frequency patterns, feature extraction is done via a discrete cosine transformation. Markov features are then generated from the transform coefficients that are retrieved.

But as advanced GANs have been developed, images are frequently manipulated using several tampering techniques at once, producing more realistic-looking images that are hard to spot as manipulated. This makes it difficult to determine the type of tampering and the precise areas of the image that are impacted. Accordingly, the hitherto successful techniques that relied solely on feature extraction are no longer able to precisely identify these alterations (Sharma et al., 2023). Deep learning techniques like Convolutional Neural Networks (CNNs) can help overcome this.

Convolutional neural networks are based on several deep artificial neural network topologies that have many hidden layers of neurones after convolutional and pooling layers. Higher-level characteristics, like images, are gradually extracted from the raw input by these layers. These techniques can mimic more complex decision functions and achieve higher classification accuracy by increasing the number of layers (Goodfellow et al., 2016; Bianchini and Scarselli, 2014). However, today's high-performing CNNs, like VGGNet (Simonyan and Zisserman, 2014), DenseNet (Huang et al., 2017), and ResNet (He et al., 2016), have a lot of layers, which makes the training process more complex and requires a lot of data because there are many local optima and a lot of hyperparameters. Furthermore, they are regarded as black-box function approximates, meaning that their judgements cannot be explained (Gu et al., 2018). Hence, this study applies CNN architecture for detection of AI generated synthetic images.

2. RESEARCH METHOD

There are two primary components to our approach. The first step is data collection and preparation, where the best photos are chosen to make up the training set after their quality is assessed. The second step is the training phase, during which a CNN network and a pre-processing pipeline are used to improve the CNN model's ability to extract and classify features. Perhaps the most significant aspect of our contribution is the use of quality measurement in the training set composition. This expands on the logical presumption that a network would concentrate on less evident features and generative process artefacts if it is trained on phoney images of high perceptual quality. As a result, it will acquire traits that are independent of the image's content, improving its capacity to identify phoney images from various ideas.

2.1 Data Acquisition

We use publicly available datasets such as FFHQ (Karras et al., 2019), AFHQ (Choi et al., 2018), and LSUN (Yu et al., 2015) to acquire genuine images. photographs of human faces may be found in the FFHQ dataset, photographs of dogs, cats, and a general class of animals can be found in the AFHQ dataset, and about one million images of ten scene categories and twenty object classes can be found in the LSUN dataset. In order to assess the cross-concept situation, we then use pretrained diffusion and GAN models to create synthetic images for various classes. In particular, we create images from pretrained networks in FFHQ, AFHQ, and LSUN-churches using the StyleGAN2 (Karras et al., 2020) model, and we create images from pretrained networks in FFHQ, LSUN-bedrooms, and LSUN-churches using the Latent Diffusion (Rombach et al., 2022).

2.2 Data Preprocessing

To guarantee the consistency and quality of the dataset used to train the CNN model, data preparation is a crucial step. In order to choose high-quality images for the training set, this step starts by evaluating the perceptual quality of both synthetic and actual images. To reduce noise and increase the learning process's dependability, low-quality photos are eliminated. After that, every image is enlarged to uniform dimensions (H×W) in order to satisfy the input requirements of the model. In order to ensure consistency and speed up convergence during training, normalisation is done by scaling pixel values to a range of [0,1]. Rotation, flipping, cropping, and colour modifications are examples of data augmentation techniques that are used to add diversity to the dataset, increasing its size and improving the model's capacity to generalise to new situations.

Balancing approaches such as undersampling the majority class or oversampling the minority class are used to resolve any imbalances between the actual and synthetic image classes. Furthermore, photos are tagged as synthetic (111) or actual (000), in accordance with the model's binary classification objective. To remove any order bias, the dataset is then randomly mixed before being divided into training, validation, and test sets in standard ratios like 70:15:15. The Synthetic Minority Over-sampling Technique with Edited Nearest Neighbours (SMOTEEN) may be used to better balance unbalanced datasets by fine-tuning oversampled samples. By ensuring that the training data is high quality, varied, and prepared, these preprocessing processes together maximise the CNN model's performance and accuracy in identifying synthetic images.

3. PROPOSED CNN MODEL FOR SYNTHETIC IMAGE DETECTION

Here, we proposed a novel CNN architecture for the identification of fake images. This method is distinguished by a complex architecture that combines two main networks via a number of intermediary processes. In particular, these networks are made up of the fully connected layers, which are essential to the classification process based on the characteristics found, and the convolutional layers, which handle the initial processing and feature extraction from the images. A more complex and precise categorisation result is made possible by this improved framework's ability to capture the nuances and complexities seen in images. A more sophisticated and useful method of visual data analysis is being heralded by the use of a CNN in the image classification job, which is an example of the convergence of cutting-edge computational methods with theoretical understanding.

The proposed CNN model's architecture is composed of several layers, each of which is intended to systematically find and examine characteristics in the input images before arriving at a fully linked layer for the classification job. Figure 1 illustrates this design.

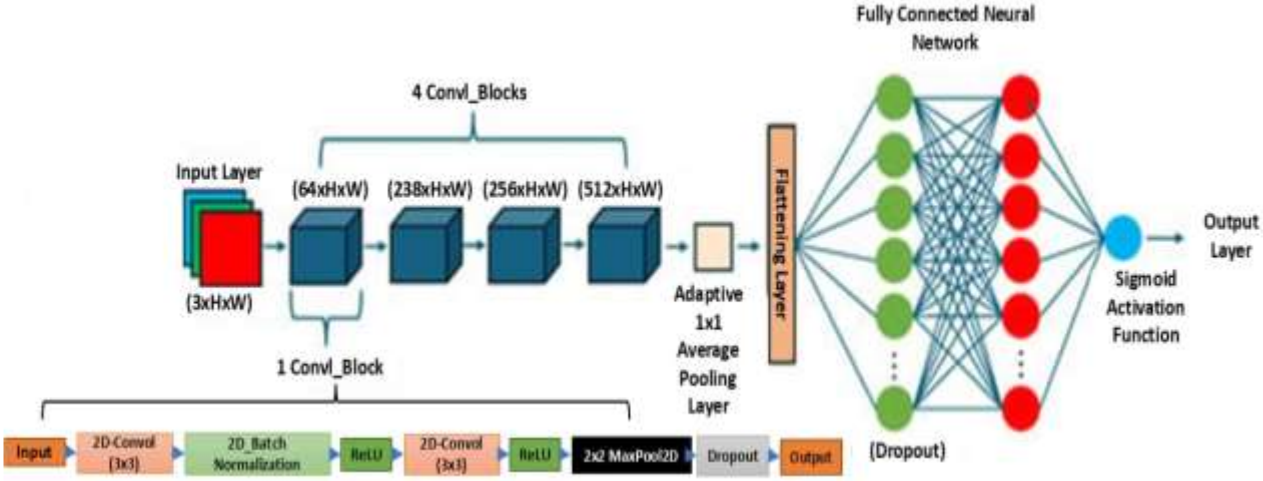

Figure 1: Architecture of the Proposed CNN Model

Images with dimensions of 3×H×W, or three colour channels across height and width, are accepted by the input layer. Each convolutional block in the design is simultaneously made up of a series of two convolutional layers, batch normalisation, and the use of ReLU activation functions. Each block's first convolutional layer uses a 3x3 kernel to extract features. While the ReLU activation function adds non-linearity, enabling the capture of intricate patterns, batch normalisation helps to stabilise the training process and speed up convergence. Additionally, because the suggested CNN design is lightweight and batch normalisation is computationally demanding (Zhu et al., 2021), stability training was ensured by employing just one of these operations per every convolutional block. In order to decrease the feature maps' spatial dimensions and computational load, max-pooling layers with a 2x2 window are employed. To combat overfitting, a dropout layer with a 0.3 rate is also used, which randomly disables a subset of the neurones during training. In order to capture more abstract and higher-level characteristics at deeper layers, the network uses four convolutional blocks to extract features, gradually increasing the number of filters (64, 128, 256, and 512). The last convolutional block is followed by an Adaptive Average Pooling layer, which shrinks each feature map to 1×1 in size and transforms the spatial dimensions into a single vector representation. A fully connected neural network with two hidden layers and 512 neurones each is then fed this vector. Dropout is performed between the fully connected layers to further reduce overfitting and improve the network's capacity for generalisation. The output layer generates a probability score for every class using a sigmoid activation function that is appropriate for binary classification problems.

3.1 Training of the CNN Model

A systematic procedure utilising the Deep Learning Toolbox is required to train the CNN model with the obtained dataset in MATLAB. The imageDatastore function allows for the efficient processing of huge image collections by loading the pre-processed dataset into MATLAB. After that, splitEachLabel is used to separate it into training, validation, and test sets. To improve variability, augmentedImageDatastore is used to supplement the data. Convolutional layers for feature extraction, batch normalisation for stability, ReLU activation for non-linearity, and fully connected layers for classification are all components of the CNN architecture, which is specified by the layerGraph and layer functions. To differentiate between actual and fake images, the last layer uses a sigmoid activation function for binary classification.

By defining the optimiser (i.e., Adam), learning rate, mini-batch size, and number of epochs, the trainingOptions function regulates the training procedure. Accuracy and loss measures are used for both training and validation as the trainNetwork function iteratively trains the model. To avoid overfitting, early stopping is used, and classification is used to assess the trained model on the test dataset. To evaluate the model's efficacy, important performance measures including accuracy, precision, recall, and F1-score are calculated. Confusion matrices and ROC curves are two examples of MATLAB's visualisation tools that help analyse findings and guarantee a reliable and effective model for identifying fake images.

4. System Implementation

In order to deploy the system in the actual world, the trained CNN model must be integrated into a working framework. In this stage, the preprocessing, training, and evaluation elements are combined to create a single system that can identify artificial images. The stable environment of MATLAB facilitates smooth integration by offering resources to optimise the pipeline from the capture of input images to classification. In order to feed actual or artificial images into the framework for analysis, the system starts with an image input module. The same procedures used during the model training phase are used for preprocessing these photos, which includes scaling, normalisation, and augmentation. The trained CNN model, which has been tuned to detect minute details and artefacts unique to synthetic images, is then applied to the pre-processed images.

The CNN model's outputs are processed by the classification module, which uses the sigmoid activation function to determine the likelihood that each image is real or artificial. The final categorisation label is determined by setting a decision threshold (e.g., 0.5). The technology also saves results for further review and logs forecasts. A Graphical User Interface (GUI) or an API may be used to improve usability by enabling users to input photos and interactively examine categorisation results. Finally, the assessment module guarantees the correctness and dependability of the deployed system. The system can adjust to new differences in synthetic image production thanks to the facilitation of continuous performance monitoring on fresh datasets. This implementation offers a reliable, effective, and user-friendly way to identify fake photos in practical situations.

5. SYSTEM RESULTS

The test dataset, which consists of both synthetic and actual photos that were not used in the training process, is used to assess the system's performance. The efficacy of the system in identifying synthetic images is evaluated by computing key metrics including accuracy, precision, recall, and F1-score. The outcomes show that the CNN model is able to generalise effectively to new data and achieves high accuracy. For example, recall and accuracy scores close to 1.0 indicate that the system detects synthetic images accurately with few false negatives and false positives.

To give a thorough analysis of categorisation performance, a confusion matrix is produced that displays the true positives, true negatives, false positives, and false negatives. This makes it easier to spot any particular problems the model could have, such identifying particular kinds of fake photos. The model's capacity to differentiate between classes at different decision thresholds is also measured by plotting a receiver operating characteristic (ROC) curve and calculating the area under the curve (AUC). The system's resilience is reinforced by its persistent high AUC. A 10-fold cross-validation technique was used to evaluate the system, guaranteeing a thorough assessment of its functionality. Using this approach, the dataset is divided into ten equal subsets, nine of which are utilised for training and one for testing. The procedure is then repeated ten times. As seen in Table 1, a reliable measure of the model's performance is obtained by averaging the outcomes from each fold.

Table 1: System Performance Result Validation

Iterations	Accuracy (%)	Precision	Recall	F1-Score
1	96.5	0.95	0.97	0.96
2	96.7	0.96	0.97	0.97
3	96.3	0.95	0.96	0.96
4	97.0	0.96	0.97	0.97
5	96.8	0.96	0.97	0.97
6	96.6	0.96	0.96	0.96
7	97.2	0.97	0.97	0.97
8	96.9	0.96	0.97	0.97
9	96.4	0.95	0.96	0.96
10	96.8	0.96	0.97	0.97
Mean	**96.7**	**0.96**	**0.97**	**0.97**

The 10-fold cross-validation findings show how reliable and successful the algorithm is at identifying fake photos. The model's outstanding ability in accurately categorising both real and synthetic images across a variety of datasets is demonstrated by its average accuracy of 96.7%.

The system can reduce false positives while keeping a high detection rate of synthetic images, as evidenced by precision and recall values of 0.96 and 0.97, respectively. These scores attest to the model's ability to make well-balanced decisions, spotting tiny generating artefacts without making needless misclassifications. Furthermore, the resilience of the model in managing a variety of difficult inputs is demonstrated by the F1-score of 0.97, which shows the good harmony between accuracy and recall. Figure 2 displays the system implementation's performance correctness. The illustration shows the accuracies attained across various implementation phases of the model.

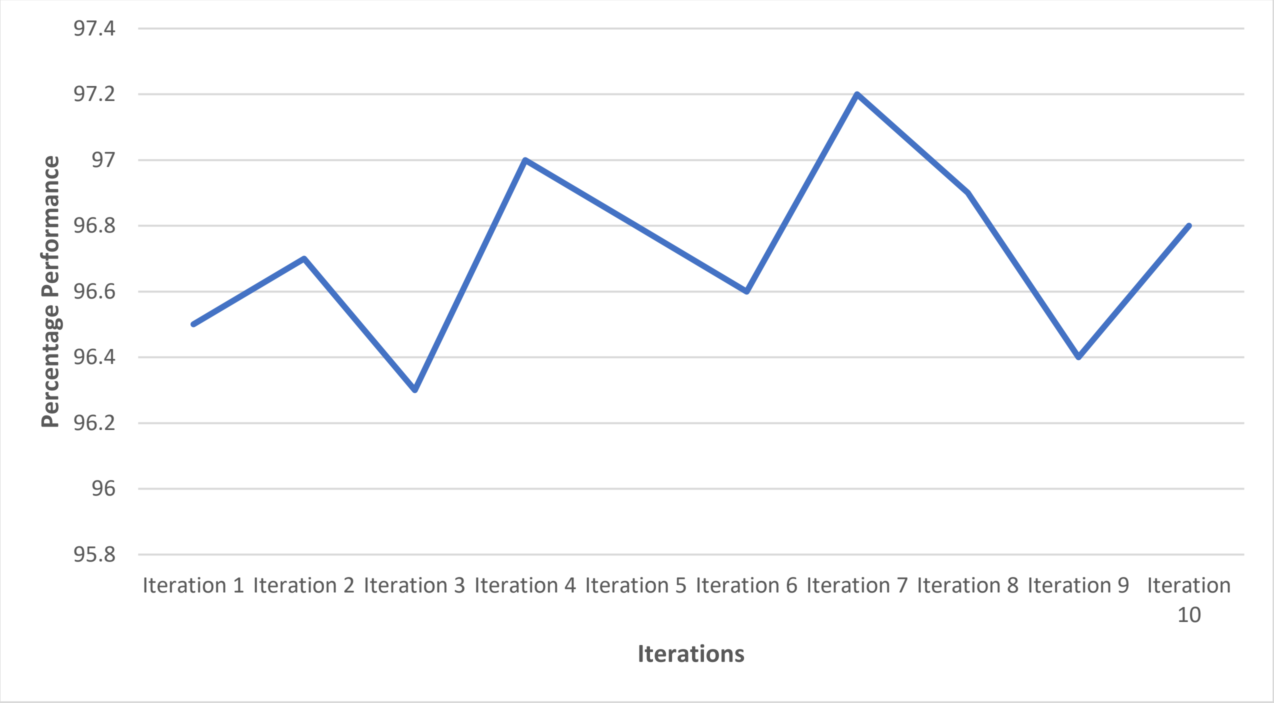

Figure 2: System Accuracy Results

The model's generalisation skills are validated by its consistent performance across all folds, which guarantees that it can adjust to changes in the dataset. For real-world applications, where the system could come across invisible synthetic image types produced by various algorithms, this resilience is essential. Even though the differences in fold results are small, they highlight how crucial cross-validation is for evaluating the model's dependability across various training and testing splits. All things considered, the system's strong accuracy, precision, recall, and F1-score metrics show that it has the potential to be a workable synthetic image detection solution that can be implemented in situations demanding sophisticated visual data processing.

6. CONCLUSION

The goal of this paper is to create a reliable system that uses a Convolutional Neural Network (CNN) to identify artificial images. Data collection and preprocessing were the first steps in the two-stage procedure, which was followed by model training and assessment. While pre-trained diffusion and GAN models were used to create synthetic images, real photographs were taken from publicly accessible databases. In order to efficiently extract characteristics from these photos and categorise them into either synthetic or actual categories, the CNN architecture was created. A 10-

fold cross-validation method was used to analyse the system, giving a comprehensive image of how well the model performed across various data subsets.

With an average accuracy of 96.7%, precision of 0.96, recall of 0.97, and F1-score of 0.97, the findings showed how successful the system was. These measures show that the model is a dependable method for synthetic image recognition since it is very good at both identifying fake images and reducing false positives. The model's resilience and generalisability are confirmed by the consistent performance throughout the ten folds. This work concludes by demonstrating how deep learning models may be used to tackle the increasing difficulty of identifying fake photos, offering a useful tool for uses including content authentication, digital forensics, and AI-generated image detection.

REFERENCES

Bayram, S., Sencar, H. T., & Memon, N. (2009). An efficient and robust method for detecting copy-move forgery. In Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 1053–1056). https://doi.org/10.1109/ICASSP.2009.4971862

Bianchini, M., & Scarselli, F. (2014). On the complexity of neural network classifiers: A comparison between shallow and deep architectures. IEEE Transactions on Neural Networks and Learning Systems, 25(9), 1553–1565. https://doi.org/10.1109/TNNLS.2013.2282551

Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., & Choo, J. (2018). StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Ferrara, P., Bianchi, T., De Rosa, A., & Piva, A. (2012). Image forgery localization via fine-grained analysis of CFA artifacts. IEEE Transactions on Information Forensics and Security, 7(5), 1566–1577. https://doi.org/10.1109/TIFS.2012.2202910

Fridrich, J., Soukal, D., & Lukás, J. (2003). Detection of copy-move forgery in digital images. International Journal of Computer Science Issues, 3(2), 55–61.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning (p. 800). MIT Press. Retrieved from http://www.deeplearningbook.org

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial networks. Advances in Neural Information Processing Systems, 3, 139–144. https://doi.org/10.5555/2993189.2993194

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., & Chen, H. (2018). Recent advances in convolutional neural networks. Pattern Recognition, 77, 354–377. https://doi.org/10.1016/j.patcog.2017.10.013

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770–778). https://doi.org/10.1109/CVPR.2016.90

He, Z., Lu, W., Sun, W., & Huang, J. (2012). Digital image splicing detection based on Markov features in DCT and DWT domain. Pattern Recognition, 45(12), 4292–4299. https://doi.org/10.1016/j.patcog.2012.05.024

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4700–4708). https://doi.org/10.1109/CVPR.2017.643

Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of StyleGAN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Li, G., Wu, Q., Tu, D., & Sun, S. (2007). A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD. In Proceedings of the 2007 IEEE International Conference on Multimedia and Expo (pp. 1750–1753). https://doi.org/10.1109/ICME.2007.4284784

Masood, M., Nawaz, M., Malik, K., Javed, A., Irtaza, A., & Malik, H. (2022). Deepfakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward. Applied Intelligence, 53(7), 3974–4026. https://doi.org/10.1007/s10462-022-09932-1

Mirsky, Y., & Lee, W. (2021). The creation and detection of deepfakes: A survey. ACM Computing Surveys, 54(9), 1–41. https://doi.org/10.1145/3433181.3433186

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Sharma, D. K., Singh, B., Agarwal, S., Garg, L., Kim, C., & Jung, K. H. (2023). A survey of detection and mitigation for fake images on social media platforms. Applied Sciences, 13(20), 10980. https://doi.org/10.3390/app132010980

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv. https://arxiv.org/abs/1409.1556

Thakur, R., & Rohilla, R. (2020). Recent advances in digital image manipulation detection techniques: A brief review. Forensic Science International, 312, 110311. https://doi.org/10.1016/j.forsciint.2020.110311

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., & Xiao, J. (2015). LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365.

Zhu, Y., Du, J., Zhu, Y., Wang, Y., Ou, Z., Feng, F., & Tang, J. (2021). Training BatchNorm only in neural architecture search and beyond. arXiv preprint arXiv:2112.00265.