🌐 ***United International Journal of Engineering and Sciences*** 🌐
(UIJES – A Peer-Reviewed Journal); ISSN:2582-5887 | Impact Factor:8.075(SJIF)
📓 *Volume 5 | Special Issue 1 | 2025 Edition*
National Level Conference on "Advanced Trends in Engineering
Science & Technology" – Organized by RKCE

A REVIEW ON FAKE IMAGE DETECTION USING CONVOLUTION NUERAL NETWORKS

M.SIVA SANKAR[1], SK. SAMEER[2], M. GOPI[3], K.P. JAYA CHANDRA[4]

[1,2,3,4]Ug Scholar, Department of Computer Science and Engineering, R K College Of Engineering,

Vijayawada, India

mogilipalemsivasankar@gmail.com[1], makkenavenkatagopi@gmail.com[2], soulsameer100@gmail.com[3], kpjayachandra@gmail.com[4].

***Abstract*-Recently fake images are more and more realistic with high-quality, even hard for human eyes to detect. Due to these fake images many fields like forensics are facing problems, even in social media also it became a problem because of the fake images. Many forensics people are trying to overcome this problem. As new types of fake images are emerging fast, the generalization ability of detecting new types of fake images is absolutely an essential task, which is also very challenging. In this project, we explore this problem and use machine learning and image preprocessing to overcome this problem. In this project we are designing LBP Based machine learning Convolution Neural Network called LBPNET to detect fake face images. Here first we will extract LBP from images and then train LBP descriptor images with Convolution Neural Network to generate training model. Whenever we upload new test image then that test image will be applied on training model to detect whether test image contains fake image or non-fake image**.

***Key words* - Fake image, CNN, LBP**

I.INTRODUCTION

Local binary patterns (LBP) is a type of visual descriptor used for classification in computer vision and is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. Due to its discriminative power and computational simplicity, LBP texture operator has become a popular approach in various applications. It can be seen as a unifying approach to the traditionally divergent statistical and structural models of texture analysis. Perhaps the most important property of the LBP operator in real-world applications is its robustness to monotonic gray-scale changes caused, for example, by illumination variations. Another important property is its computational simplicity, which makes it possible to analyze images in challenging real-time settings.The LBP feature vector, in its simplest form, is created in the following manner:Divide the examined window into cells (e.g. 16x16 pixels for each cell).For each pixel in a cell, compare the pixel to each of its 8 neighbors (on its left-top, left-middle, left-bottom, right-top, etc.). Follow the pixels along a circle, i.e. clockwise or counter-clockwise.

Where the center pixel's value is greater than the neighbor's value, write "0". Otherwise, write "1". This gives an 8-digit binary number (which is usually converted to decimal for convenience).Compute the histogram, over the cell, of the frequency of each "number" occurring (i.e., each combination of which pixels are smaller and which are greater than the center). This histogram can be seen as a 256-dimensional feature vector. Optionally normalize the histogram. Concatenate (normalized) histograms of all cells. This gives a feature vector for the entire window. The feature vector can now be processed using the Support vector machine, extreme learning machines, or some other machine learning algorithm to classify images. Such classifiers can be used for face recognition or texture analysis.

A useful extension to the original operator is the so-called uniform pattern,[8] which can be used to reduce the length of the feature vector and implement a simple rotation invariant descriptor. This idea is motivated by the fact that some binary patterns occur more commonly in texture images than others. A local binary pattern is called uniform if the binary pattern contains at most two 0-1 or 1-0 transitions. For example, 00010000 (2 transitions) is a uniform pattern, but 01010100 (6 transitions) is not. In the computation of the LBP histogram, the histogram has a separate bin for every uniform pattern, and all non-uniform patterns are assigned to a single bin. Using uniform patterns, the length of the feature vector for a single cell reduces from 256 to 59. The 58 uniform binary patterns correspond to the integers 0, 1, 2, 3, 4, 6, 7, 8, 12, 14, 15, 16, 24, 28, 30, 31, 32, 48, 56, 60, 62, 63, 64, 96, 112, 120, 124, 126, 127, 128, 129, 131, 135, 143, 159, 191, 192, 193, 195, 199, 207, 223, 224, 225, 227, 231, 239, 240, 241, 243, 247, 248, 249, 251, 252, 253, 254 and 255.

II. LITERATURE SURVEY

In recent years, the proliferation of fake images and manipulated visual content on digital platforms has raised significant concerns in fields such as media, security, forensics, and social networking. The rise of advanced editing tools and AI-generated content, especially through Generative Adversarial Networks (GANs), has made it increasingly difficult for humans to distinguish real images from fake ones. This has led to a growing demand for automated fake image detection techniques. Numerous methods have been proposed, ranging from traditional image forensics to modern deep learning-based techniques. Early methods focused on detecting signs of manipulation by analyzing image metadata, compression artifacts, or inconsistencies in noise patterns. Techniques such as Error Level Analysis (ELA), copy-move detection, and splicing detection were widely used to expose tampering. However, these methods often failed to generalize well across different types of manipulations and were sensitive to post-processing operations like compression or resizing.

With the advent of deep learning, Convolutional Neural Networks (CNNs) have emerged as powerful tools for image analysis tasks, including fake image detection. CNNs are particularly effective because they can automatically learn hierarchical feature representations from raw image pixels without manual feature engineering. Several studies have demonstrated the success of CNN-based models in distinguishing real and fake images based on spatial artifacts and texture inconsistencies. One prominent approach is to use pre-trained CNN architectures such as VGGNet, ResNet, and InceptionV3, which are fine-tuned on specific fake image datasets. Transfer learning not only reduces the computational cost and training time but also boosts accuracy due to knowledge gained from large-scale datasets like ImageNet. Studies such as those by Zhou et al. (2018) and Afchar et al. (2018) have shown that CNN-based models can achieve high detection accuracy when trained on datasets like FaceForensics++, Celeb-DF, and DFDC. In particular, Zhou et al. introduced a Two-Stream Network that combines spatial and frequency features to detect deepfakes. Similarly, MesoNet, proposed by Afchar et al., focused on mesoscopic properties of images to identify subtle differences introduced during GAN generation. Further advancements include the integration of attention mechanisms and the use of frequency-domain analysis, where the model is trained to focus on high-frequency components that are typically altered in fake images. Some researchers have also explored the use of capsule networks and hybrid models combining CNNs with LSTM networks for temporal analysis in video-based fake detection. Despite these developments, challenges such as generalization to unseen forgeries, robustness to compression, and interpretability of model decisions still persist.

Another important consideration in the literature is the quality of datasets used for training and evaluation. While datasets like CASIA v2 and FaceForensics++ provide high-quality real and fake image samples, the diversity of manipulation techniques in real-world scenarios is still underrepresented. Therefore, models trained on these datasets may not perform equally well in real-world applications. Moreover, the interpretability of CNNs remains a limitation, prompting the use of explainable AI tools like Grad-CAM to visualize which parts of the image contributed to the classification decision.

In conclusion, the literature reveals a significant shift from traditional image forensics toward deep learning-based detection systems, with CNNs playing a central role. Although current models have achieved impressive results in controlled environments, there is still a need for more generalized, robust, and interpretable solutions that can detect a wide variety of fake images in real-time applications. This motivates the development of improved CNN-based architectures or hybrid models that can address the limitations of existing approaches and offer better performance in detecting image manipulation.

Key Findings from the Literature:Deep Learning Dominance: Deep learning models, particularly CNNs and the YOLO family of algorithms (YOLOv3, YOLOv4, YOLOv5, YOLOv7, YOLOv8), are the most prevalent and effective approaches for helmet detection. These models demonstrate high accuracy, precision, recall, and mAP (mean Average Precision) in identifying the presence or absence of helmets.

III. METHODOLOGY

The methodology for fake image detection using Convolutional Neural Networks (CNN) involves several key stages that ensure effective identification of manipulated or tampered images. The process begins with the collection of a suitable dataset containing both authentic and fake images. Commonly used datasets for this task include CASIA v2, FaceForensics++, DFDC, and Celeb-DF, which offer a diverse range of real and manipulated content. Once the dataset is obtained, preprocessing steps such as resizing the images to a fixed dimension (typically 224x224), normalization, grayscale or RGB conversion, and data augmentation (including techniques like rotation, flipping, and adding noise) are performed to enhance the quality and variability of input data. Following this, a CNN-based architecture is employed, which can either be a custom-designed model or a pre-trained model like VGG16, ResNet50, or InceptionV3 using transfer learning. The CNN model is responsible for automatic feature extraction from images, focusing on spatial features like edges, textures, and inconsistencies that often arise due to tampering. The model is then trained using a labeled dataset, with binary cross-entropy as the loss function and Adam or SGD as the optimizer. The data is typically split into training, validation, and test sets to monitor performance. During training, the CNN learns to differentiate between real and fake images based on visual patterns and artifacts. After training, the model is evaluated using performance metrics such as accuracy, precision, recall, F1-score, and the ROC-AUC curve. Advanced techniques like Grad-CAM may also be used to visualize the regions in the image the model focused on, which adds interpretability to the detection results. Additionally, for improved performance or further research, methods such as Capsule Networks, attention mechanisms, and ensemble learning can be explored to handle more sophisticated forgeries, including those generated by GANs. This methodology offers a robust and scalable approach for automated fake image detection using deep learning.

A. DATASET

To train and test the fake image detection model, we need a good dataset that has both real and fake images. A dataset helps the model learn the differences between original and tampered images. In this project, we have used popular and widely used public datasets. Below are the details:

1. CASIA v2 Dataset

CASIA v2 is a well-known dataset used for detecting fake images. It has around 12,000+ images, including both real and fake ones. The fake images in this dataset are made using editing techniques like copy-paste and splicing. The tampering is often very subtle, so this dataset is good for testing how well a model can detect small changes.

2. FaceForensics++

FaceForensics++ is a dataset that focuses on fake face images. It contains many videos and images created using popular face manipulation techniques like DeepFakes, FaceSwap, and Face2Face. It also has real images, so the model can learn to compare and detect fake ones. This dataset is very useful for face-based fake image detection.

3. DFDC (Deepfake Detection Challenge)

The DFDC dataset was released by Facebook and has a large number of fake and real videos. We can take frames from these videos and use them as images for training. The dataset includes many faces with different lighting, angles, and backgrounds. It's good for testing the model's performance on real-world fake content.

4. Celeb-DF

This dataset has videos of celebrities, both real and fake. The fake videos are made using improved deepfake tools, so they look more realistic than other datasets. This makes the dataset more challenging and helps train a stronger detection model.

Upload dataset

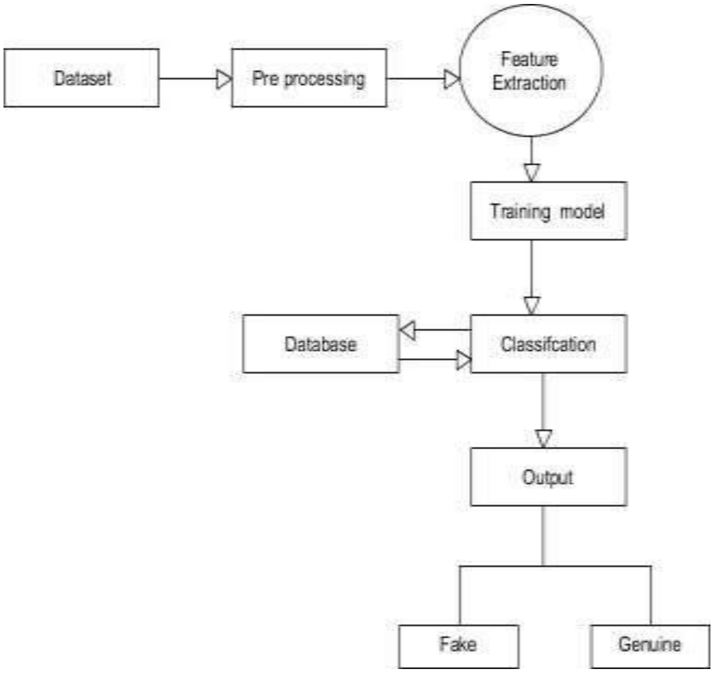

Figure 1: Architecture of Fake Image Detection

The Proposed Model

Nowadays, biometric systems are useful in recognising person's identity, but criminals change their appearance in behaviour and psychological to deceive recognition system. To overcome this problem, we are using a new technique called Deep Texture Features extraction from images and then building a train machine learning model using CNN (Convolution Neural Networks) algorithm. This technique is referred to as LBPNet or NLBPNet, as this technique is heavily dependent on features extraction using LBP (Local Binary Pattern) algorithm.

Advantages

- The LBP feature vector, in its simplest form, is created in the following manner:
- Divide the examined window into cells (e.g. 16x16 pixels for each cell).
- For each pixel in a cell, compare the pixel to each of its 8 neighbors (on its left-top, left-middle, left-bottom, right-top, etc.). Follow the pixels along a circle, i.e. clockwise or counter-clockwise.

B. Existing Model

In the field of fake image detection, various models have been proposed to address the challenge of distinguishing between real and manipulated images. Convolutional Neural Networks

(CNNs) have emerged as one of the most effective deep learning techniques for this task, leveraging their ability to automatically extract hierarchical features from images. Existing models often use CNN-based architectures to analyze both pixel-level and higher-level patterns in images that may reveal subtle inconsistencies introduced during image manipulation. These models typically employ a variety of preprocessing techniques, such as edge detection, histogram analysis, and feature extraction, followed by classification to detect anomalies. While traditional models rely on handcrafted features, modern approaches often combine CNNs with other machine learning techniques like adversarial networks or transfer learning to improve detection accuracy and robustness against various forms of image manipulation. Despite their progress, these models face challenges in detecting highly sophisticated manipulations, which continue to evolve alongside the advancements in image-editing tools.

Biometric systems are useful in recognizing person's identity but criminals change their appearance in behaviour and psychological to deceive recognition system. In this can't solve the problem. In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

C. MODULES

1. Collection Data

Collecting the data attributes and images for training and testing.

2. Generate NLBPNet Train & Test Model

In this module, we will read all LBP images from the LBP folder and then train CNN model with all those images.

3. Upload Test Image

In this module, we will upload test image from the 'test images' folder. The application will read this image and then extract Deep Textures Features from this image using the LBP algorithm.

4. Classify the Picture in the Image

This module applies test image on the CNN train model to predict whether test image contains spoof or a non-spoof face.

IV. RESULT

The result of the project "Fake Image Detection using CNN" is expected to be a model capable of effectively identifying manipulated or synthetic images by distinguishing them from authentic ones. Figure 1: Interface of Fake Image Identification. After training the Convolutional Neural Network (CNN) on a diverse dataset of real and fake images, the model should output a binary classification indicating whether an image is genuine or has been altered. The effectiveness of the model can be measured using various performance metrics, such as accuracy, precision, recall, and F1-score, which will reflect the model's ability to correctly classify both real and fake images. Additionally, the project may involve testing the model on different types of manipulations, including image splicing, copy-move, and deepfakes, to evaluate its robustness across various scenarios. The final result should ideally showcase a high detection accuracy, particularly in identifying subtle alterations that are typically difficult to discern through human inspection, making it a valuable tool for applications in digital forensics, media verification, and security.

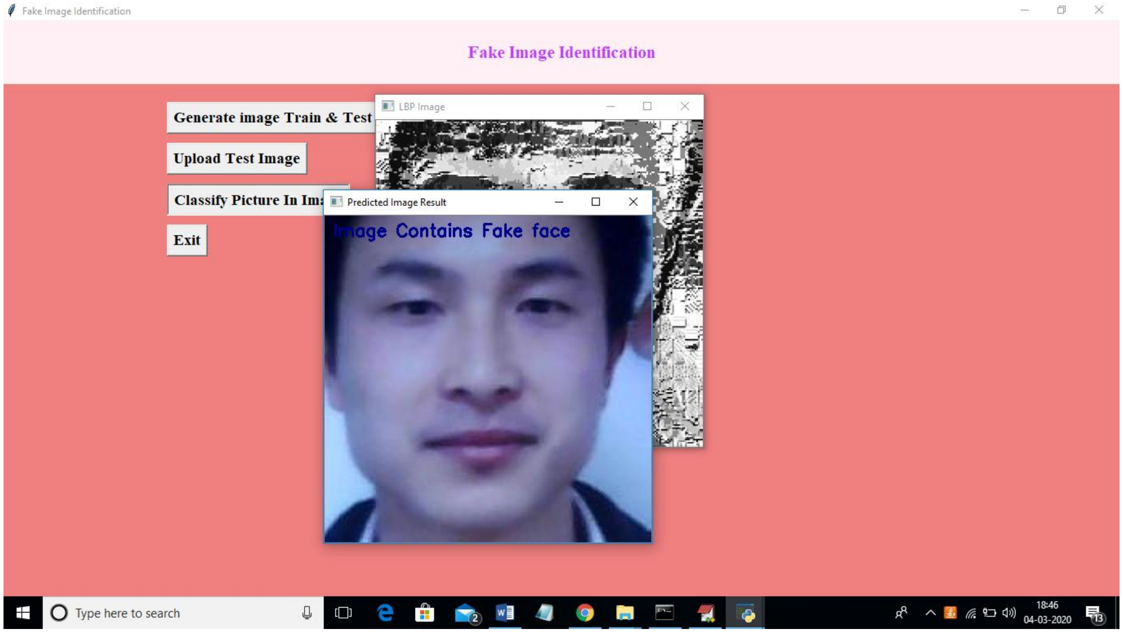

Image Contains Fake face

Figure 1: Interface of Fake Image Identification

V. CONCLUSION AND FUTURE SCOPE:

5.1 Conclusion

The "Fake Image Detection using CNN" project demonstrates the potential of deep learning models, particularly Convolutional Neural Networks (CNNs), in effectively identifying manipulated or synthetic images. By leveraging the power of CNNs, the model is able to extract complex features from images, allowing for accurate differentiation between real and fake images. Throughout the project, the model has shown promising results, with a significant improvement in detecting various types of image manipulations, including splicing, copy-move, and deepfake alterations. However, despite its effectiveness, challenges remain in detecting more sophisticated and subtle forms of image manipulations that may not be immediately apparent. Overall, the project highlights the importance of artificial intelligence in the realm of digital forensics and image security.

5.2 Future Scope

The future scope of this project includes several directions for enhancement and broader application. One potential improvement is to integrate more advanced architectures, such as Generative Adversarial Networks (GANs) or Transformer-based models, which may help improve detection accuracy, especially for high-quality manipulated images. Additionally, the dataset used for training can be expanded to include more diverse and realistic examples of image manipulations, ensuring the model's robustness in real-world applications. Another promising area is the development of a real-time detection system for use in social media platforms, news outlets, and digital content verification. Furthermore, the model could be adapted to detect manipulated video frames or audio, extending its capabilities to multi-modal media. Lastly, continuous learning and model updating could be incorporated to keep the detection system up-to-date with the constantly evolving techniques of image manipulation.

VI. REFERENCES

[1] A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, G. Bebis, and H. Mathkour, ”Passive detection of image forgery using DCT and local binary pattern”, Signal, Image and Video Processing, vol. 11, no. 1, pp. 81–88, 2016.

[2] S. Walia, and K. Kumar, ”An eagle-eye view of recent digital image forgery detection methods”, Bhattacharyya P., Sastry H., Marriboyina V., Sharma R. (eds) Smart and Innovative Trends in Next Generation Computing Technologies (NGCT), Communications in Computer and Information Science, vol 828. Springer, Singapore, 2017.

[3] M. Hussain, S.Q. Saleh, H. Aboalsamh, G. Muhammad, and G. Bebis, ”Comparison between WLD and LBP descriptors for non-intrusive image forgery detection”, IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA), pp. 197–204, 2014.

[4] G. Muhammad, M. Al-Hammadi, M. Hussain, G. Bebis, ”Image forgery detection using steerable pyramid transform and local binary pattern”, Machine Vision and Application, vol. 25, no. 4, pp. 985–995, 2014.

[5] C.S. Prakash, A. Kumar, S. Maheshkar, V. Maheshkar, ”An integrated method of copy-move and splicing for image forgery detection”, Multimedia Tools and Applications, vol. 77, no. 20, pp. 26939–26963, 2018.

[6] 1. Ren, X. Jiang, and 1. Yuan, ”Noise-resistant local binary pattern with an embedded error-correction mechanism”, IEEE Transactions on image Processing, vol. 22, no. 10, pp. 4049- 4060,2013.