ISSN NO: 9726-001X

Volume 13 Issue 02 2025

JOURNAL OF CURRENT SCIENCE

DEEP LEARNING-BASED APPROACH FOR DETECTING COPY -MOVE FORGERY

KODI MURARI,
UG Student,
Department of CSE,
St. Martin's Engineering College,
Secunderabad, Telangana, India
murarikodi@gmail.com

C. YOSEPU,
Assistant Professor,
Department of CSE,
St. Martin's Engineering College,
Secunderabad, Telangana, India
yosepucse@smec.ac.in

Abstract- Copy-move forgery is a prevalent form of image manipulation where a part of an image is duplicated and pasted onto another area, often to conceal or alter information. It's commonly encountered in digital media forensics, with applications in detecting tampered images, verifying authenticity, and ensuring integrity in legal and journalistic contexts. Currently, detecting copy-move forgery relies heavily on manual analysis by forensic experts. This process involves visually inspecting images, looking for inconsistencies in textures, lighting, and patterns. Despite its reliability, manual analysis is time-consuming and resource-intensive, limiting its scalability and efficiency in handling large datasets. Manual analysis suffers from subjectivity and human error, leading to potential inaccuracies in identifying forged regions. It's impractical for processing a vast number of images quickly, hindering its applicability in real-time scenarios. Additionally, the increasing sophistication of forgery techniques demands more robust and automated solutions. VGG 16, a convolutional neural network (CNN), offers a promising solution for automating copy-move forgery detection. Trained on extensive datasets, VGG 16 excels in feature extraction, enabling it to recognize patterns indicative of tampering with high accuracy. Its hierarchical architecture allows for the detection of both global and local inconsistencies in images, enhancing its versatility and effectiveness in identifying forged regions.

Keywords: Copy-Move Forgery, Image Manipulation, Digital Media Forensics, Tampered Images, Authenticity Verification, Manual Analysis, Forensic Experts, Human Error, Real-Time Processing, Forgery Techniques, VGG 16, Convolutional Neural Network (CNN), Feature Extraction, Pattern Recognition.

I. INTRODUCTION

Copy-move forgery presents a significant challenge in digital image forensics, involving the duplication and relocation of image sections within the same file. Such manipulations can alter evidence in legal cases or mislead viewers in journalism. Detecting these subtle changes is complex, as copied regions often blend seamlessly with their surroundings. Traditional detection methods rely on manual inspection by forensic experts, who assess inconsistencies in textures, lighting, and patterns. While reliable, this approach is time-consuming, prone to human error, and limited in scalability, especially with the growing volume of digital content. Thus, automated solutions capable of real-time detection are essential. Deep learning, particularly convolutional neural networks (CNNs) like VGG 16, offers a promising avenue for identifying tampering through enhanced feature extraction and pattern recognition.

Detecting copy-move forgery is critical in fields such as law enforcement and journalism. Current detection methods are mainly manual, requiring forensic analysts to identify irregularities in images. This process is labor-intensive, slow, and vulnerable to human error, especially as forgery techniques become more sophisticated, seamlessly blending copied regions into the image. The increasing volume of digital images necessitates scalable solutions for efficient, real-time analysis. The absence of automated detection methods poses significant risks, including the unchecked circulation of manipulated images that can lead to misinformation and legal issues. Therefore, there is an urgent need for robust automated systems capable of high-accuracy detection.

II. RELATED WORK

[1] K. Lalli (2023) Digital image manipulation has become prevalent with the rise of software tools and mobile apps, leading to significant concerns about the spread of forged images on social media. Common forms of forgery include copy-move forgery, where parts of an image are duplicated, and image splicing, where segments from different images are combined. This proliferation of tampered images erodes trust in digital content.

[2] Meet Patel (2023) This paper presents a novel image forgery detection system using Convolutional Neural Networks (CNNs) to identify various manipulations, including copy-move, splicing, and retouching. The system combines Error Level Analysis (ELA) with deep learning techniques to enhance accuracy and reliability. Evaluated on a dataset of real-world images, it achieved a detection accuracy of 93%, outperforming existing methods.

[3] Dipanshu Narayan (2023) Images shared online are often altered, with manipulations like compression and resizing complicating forgery detection. One prevalent type of image fraud is copy-move forgery, where a section of an image is duplicated and placed elsewhere, making it challenging to detect due to the similarity of attributes. This study presents a method for identifying copy-move forgeries by processing image blocks into features using transforms.

[4] Parita Mer (2023) The detection of image forgeries is a vital area in digital image analysis, focused on identifying and locating modified regions within images. With the rise of advanced image editing tools and increasingly complex forgery techniques, the need for robust detection methods has become more critical. This survey paper provides a comprehensive classification of forgery detection methods, considering factors such as types of forgery, detection methodologies, evaluation metrics, and datasets used.

[8] Gul Muzaffer (2019) Copy-move forgery, which involves duplicating or removing objects in images, is easily executed using manipulation programs. Traditional detection methods include block-based and keypoint-based approaches using hand-crafted features. This paper introduces a new deep learning-based forgery detection scheme utilizing the pre-trained AlexNet model to extract feature vectors from overlapping image subblocks. The results indicate a higher accuracy rate compared to traditional methods reported in the literature.

[9] Arfa Binti Zainal Abidin (2019) The rise of user-friendly image editing software has made digital image manipulation increasingly accessible, leading to a surge in forged images used for malicious purposes, such as spreading fake news. As tools like Adobe Photoshop and Pixir become more sophisticated, distinguishing between manipulated and authentic images becomes challenging. This has sparked significant interest in digital image forensics, prompting researchers to develop various forgery detection techniques.

[10] Rahul Thakur (2019) The rapid advancements in digital image processing have led to a surge in doctored images created using software like GNU Gimp and Adobe Photoshop, raising concerns in sectors such as news, politics, and entertainment. This highlights the urgent need for effective image tampering detection systems to differentiate between authentic and manipulated images. Common tampering techniques include copy-move forgery and splicing, which present significant challenges for detection.

III. PROPOSED WORK

3.1 Overview

Step 1:

Copy-move forgery is a straightforward technique that manipulates digital images by copying and pasting portions within the same image, potentially concealing or duplicating objects. This tampering poses risks in journalism, legal investigations, and scientific research where image authenticity is vital. Manual detection is labor-intensive and prone to human error. To tackle these issues, we propose a deep learning-based method using the VGG16 convolutional neural network to automate copy-move forgery detection.

Step 2: Dataset and Image Preprocessing

We use the MICC-F220 dataset, which includes both authentic ('AU') and tampered ('TU') images. Images are resized to 64x64 pixels, converted to RGB, and normalized to pixel values between 0 and 1. This standardization aids the neural network’s learning process. The dataset is shuffled and split into 80% for training and 20% for testing.

Step 3: Existing System

Traditionally, copy-move forgery detection relies on expert manual inspection, identifying anomalies in texture and lighting. While reliable, this approach is time-consuming and subjective, leading to potential errors. The sophistication of forgery techniques necessitates automated solutions for efficient and accurate detection.

Step 4: Clustering with DBSCAN

To enhance forgery detection, we apply the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm for image segmentation. DBSCAN effectively identifies clusters of varying shapes and sizes, making it suitable for spotting irregular patterns associated with forgery. We cluster superpixels, which are segments of the image with similar properties, to isolate areas exhibiting unusual patterns indicative of tampering.

Step 5: Proposed CNN with VGG16

Our approach centers on the VGG16 convolutional neural network, known for its performance in image recognition. Pretrained on ImageNet, VGG16 effectively extracts high-level features. We modify the model by removing the top layers and adding custom layers, including average pooling, flattening, and dense layers with ReLU and softmax activations for classification. The model uses the Adam optimizer and binary cross-entropy loss function, achieving high accuracy. We save the best model based on validation accuracy for better generalization.

Step 6: Performance Comparison

We evaluate our method against traditional manual analysis and other automated techniques. The VGG16-based model shows superior accuracy in detecting copy-move forgery, significantly reducing the time and effort of manual inspection. Our model achieves high precision, recall, and F1-score, effectively identifying both forged and authentic images.

The confusion matrix generated from the model's predictions further highlights its robustness, with minimal misclassifications. The performance metrics confirm that our deep learning-based approach offers a reliable and scalable solution for copy-move forgery detection.

Fig.4.1.1: Architectural Block Diagram of The Proposed system.

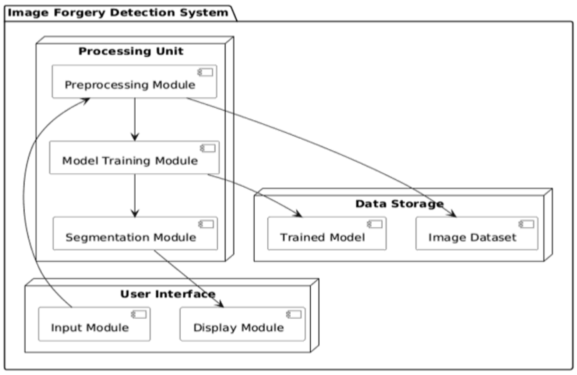


FIG. 3.1: Architectural Block Diagram of Proposed System

3.2 Image Preprocessing

Image preprocessing is vital for computer vision and image analysis, preparing raw images for algorithms or neural networks. Key steps include:

Step 1. Image Read

The first step involves reading the raw image from a source, typically a file on disk. Supported formats include JPEG, PNG, and BMP. Libraries specific to the programming environment facilitate this process, producing a digital representation that can be manipulated programmatically.

Step 2. Image Resize

Resizing is a common preprocessing step, especially for machine learning models and deep neural networks. It alters the image's dimensions and is crucial for several reasons:

- Ensuring uniform input size: Many machine learning models, particularly CNNs, require consistent image dimensions, and resizing standardizes these sizes.
- Reducing computational complexity: Smaller images demand fewer computations, facilitating faster training and inference.
- Managing memory constraints: Resizing may be necessary to fit images within available memory limits.
- When resizing, it's essential to maintain the aspect ratio to prevent image distortion. Typically, libraries like OpenCV or Pillow provide convenient functions for resizing images.

Step 3. Image to Array

In this step, the image is converted into a numerical representation as a multidimensional array or tensor. Each pixel corresponds to a value in the array, structured with dimensions for height, width, and color channels. Grayscale images yield a 2D array representing pixel intensity, while color images result in a 3D or 4D array that includes color channels (e.g., RGB) and possibly batch size. This conversion enables numerical manipulation and analysis, making the data compatible with libraries like NumPy or TensorFlow.

Step 4. Image to Float32

Most machine learning algorithms expect input data as 32-bit floating-point numbers (float32). Converting the image array to float32 ensures pixel values can represent a range between 0.0 (black) and 1.0 (white) or -1.0 and 1.0, depending on normalization. This step maintains consistency in data types and compatibility with machine learning frameworks, typically achieved by dividing pixel values by the maximum intensity (e.g., 255 for 8-bit images).

Step 5. Image to Binary

Image binarization converts a grayscale image into a binary image, where each pixel is represented as 0 (black) or 1 (white) based on a threshold. This process is often used in image segmentation to separate objects from the background. By setting a threshold, pixels above it are set to 1, while those below are set to 0. Binarization simplifies the image, isolating essential information for applications like character recognition or object tracking**.**

3.3 Dataset Splitting

In machine learning, dividing the dataset into training and test sets is crucial for model performance. Training on one dataset and testing on another helps the model learn correlations effectively. A well-trained model should perform well on both training and test sets. The definitions are as follows:

Training Set: A subset used to train the model, where the output is known.

Test Set: A subset used to evaluate the model's predictions.

3.4 CNN Model

Convolutional Neural Networks (CNNs) are powerful for image classification, automatically learning features from raw data rather than relying on handcrafted features. This section explores CNN architecture, training methodologies, and applications**.**

3.4.1 CNN Layers

In a CNN for larvae image classification, the architecture consists of several layers, each focused on feature extraction and prediction. Key components include:

Input Layer: Receives raw pixel values based on input image dimensions (width, height, channels). Convolutional Layers: Core building blocks that use multiple filters (kernels) sliding over the input image to produce feature maps. These filters capture local patterns and spatial relationships, allowing the network to learn hierarchical feature representations. During the forward pass, each filter computes element-wise multiplications and summations, storing results in a 2D activation map.

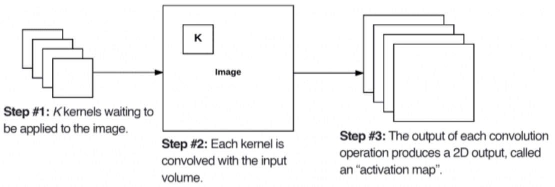

K
Image
Step #1: K kernels waiting to be applied to the image.
Step #2: Each kernel is convolved with the input volume.
Step #3: The output of each convolution operation produces a 2D output, called an "activation map".

FIG 3.2 The Pipeline of the General CNN Architecture

After applying all K filters to the input volume, we now have K, 2-dimensional activation maps. We then stack our K activation maps along the depth dimension of our array to form the final output volume

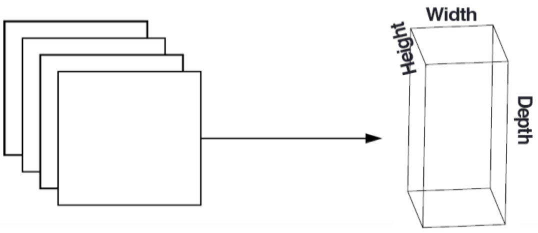


FIG 3.3 Stacking of Activation map

Figure 2: After obtaining the K activation maps, they are stacked together to form the input volume to the next layer in the network.

Activation Function:

Typically, each convolutional layer is followed by an activation function such as ReLU (Rectified Linear Unit). The activation function introduces non-linearity into the network, enabling it to learn complex relationships between features.

ReLU (Rectified Linear Unit) Activation Function

The ReLU is the most used activation function in the world right now.Since, it is used in almost all the convolutional neural networks or deep learning.

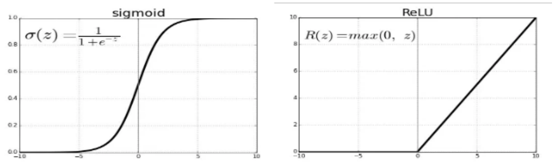


FIG 4.4 Activation function used between the hidden layers

As seen, the ReLU activation function is half rectified: f(z)f(z) is zero for z<0z<0 and equal to zz for z≥0z≥0, with a range of [0,∞)[0,∞). Both the function and its derivative are monotonic. However, the immediate zeroing of negative values can hinder the model's ability to learn effectively, as negative inputs are not appropriately mapped, impacting the resulting graph.

Pooling Layers

Pooling layers reduce the spatial dimensions of feature maps while preserving important information. For instance, max pooling selects the maximum value from a region of the feature map, effectively downsampling it.

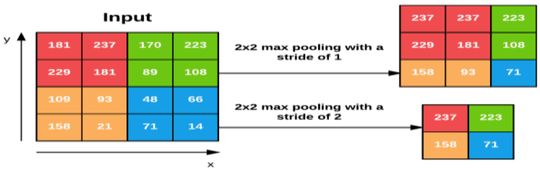


FIG 4.5 Max pooling Layer

We can further reduce the output volume size by increasing the stride. By applying a stride of S=2S=2, we keep only the largest value from each 2×2 block in the input and then step two pixels over to repeat the operation. This pooling reduces the width and height by half, effectively discarding 75% of the activations from the previous layer.

In summary, pooling layers accept an input volume of size Winput×Hinput×DinputWinput×Hinput×Dinput and require two parameters:

- Receptive Field Size FF (pool size)
- Stride S

The output volume size is given by:
Woutput = ((Winput −F) / S) +1
Houtput = ((Hinput −F) / S) +1
Doutput = Dinput

Fully Connected Layers:

After several convolutional and pooling layers, the feature maps are flattened into a vector and passed through one or more fully connected (dense) layers, which perform high-level reasoning and decision-making based on the extracted features.
Output Layer

The output layer of the CNN generates the final predictions. In larvae image classification, it typically consists of one neuron per class, using a softmax activation function to produce probabilities for each class.

3.4.2 Loss Function

During training, the CNN computes a loss function to measure the difference between predicted probabilities and actual labels. Common loss functions for classification include categorical and binary cross-entropy. The loss function quantifies how well the network models the training data, and the goal is to minimize this loss. Hyperparameters are adjusted to minimize the average loss, identifying the weights wTwT and biases bb that minimize JJ (average loss).

$$J(w^T, b) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)})$$

3.5 CNN VGG16

A Convolutional Neural Network (CNN) is a deep learning model tailored for processing structured grid-like data, such as images. It comprises multiple layers, including convolutional, pooling, and fully connected layers, making CNNs highly effective for image classification, object detection, and segmentation due to their hierarchical feature extraction.

VGG-16

The VGG-16 model, developed by the Visual Geometry Group at the University of Oxford, features 16 layers, including 13 convolutional and 3 fully connected layers. Known for its simplicity and effectiveness, VGG-16 excels in various computer vision tasks, including image classification and object recognition. Its architecture consists of a stack of convolutional layers followed by max-pooling layers, allowing the model to learn intricate hierarchical representations of visual features for robust predictions.

Despite its relative simplicity compared to newer architectures, VGG-16 remains a popular choice in deep learning due to its versatility and strong performance. The model gained recognition in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), achieving top ranks in tasks such as object localization and image classification, detecting objects from 200 classes and classifying images into 1,000 categories.

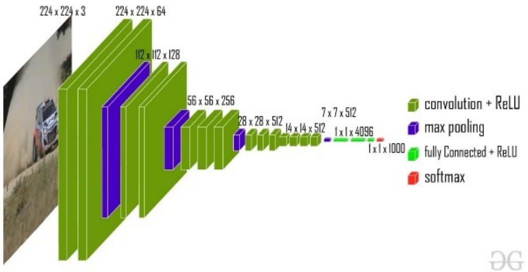


FIG 4.11 VGG-16 architecture

This model achieves 92.7% top-5 test accuracy on the ImageNet dataset, which contains 14 million images across 1,000 classes.
VGG-16 Model Objective

3.6 ADVANTAGES OF CNN

Hierarchical Feature Learning: CNNs automatically learn hierarchical representations, starting with low-level features like edges and progressing to more complex ones. This allows effective capture of intricate patterns in data.

Translation Invariance: Through convolution and pooling, CNNs recognize patterns regardless of their location, enhancing robustness to various transformations in input data.

Parameter Sharing: Weight sharing across spatial locations reduces the number of parameters, enabling CNNs to efficiently learn from large datasets and generalize well without overfitting.

Sparse Connectivity: CNNs connect neurons to small input regions, minimizing computational load and memory requirements while allowing efficient processing of high-dimensional data.

Local Receptive Fields: By focusing on small regions, CNNs capture spatial dependencies and local patterns, crucial for tasks like object recognition and segmentation.

State-of-the-Art Performance: CNNs excel in computer vision tasks such as image classification and object detection, proving indispensable for many applications.

IV. RESULTS & DISCUSSIONS

Copy-move forgery is a prevalent image manipulation technique where part of an image is copied and pasted within itself. Detecting such forgeries is crucial in digital forensics for maintaining the authenticity of images in sensitive contexts. Traditional manual detection methods are time-consuming and error-prone. To improve this, an automated approach using the VGG16 convolutional neural network (CNN) is proposed for high-accuracy copy-move forgery detection.

The VGG16 model, pre-trained on the ImageNet dataset, excels in feature extraction through its hierarchical architecture, recognizing both global and local patterns indicative of tampering. The implementation involves several key steps:

Dataset Preparation

The MICC-F220 dataset, containing both authentic and tampered images, is used. Images are resized to 64x64 pixels and converted to RGB format, with processed images and labels saved as NumPy arrays.

Image Preprocessing

Images are normalized to a scale of 0 to 1 and split into training (80%) and testing (20%) sets, ensuring diversity for model training.

Model Training

The VGG16 model is modified by adding layers suited for forgery detection, including pooling, flattening, dense, and dropout layers. The model is compiled using the Adam optimizer with a learning rate of 0.01 and trained for 30 epochs.

Forgery Detection

The model predicts forgery likelihood in test images. A modified DBSCAN algorithm segments images into super pixels, which the VGG16 model analyzes for forgery detection. SIFT (Scale-Invariant Feature

Transform) extracts key points to locate forged regions, marked visually.

Evaluation

Dataset

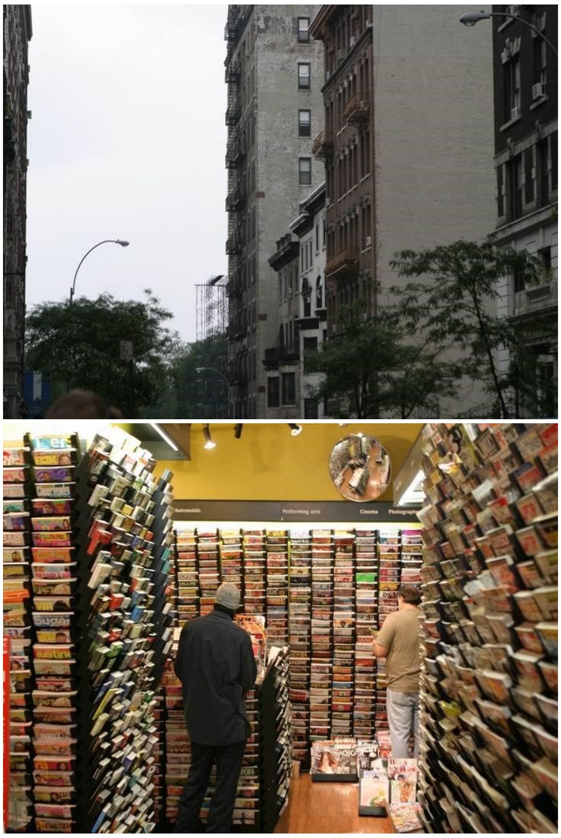


FIG 4.1: : Normal Images (AU)

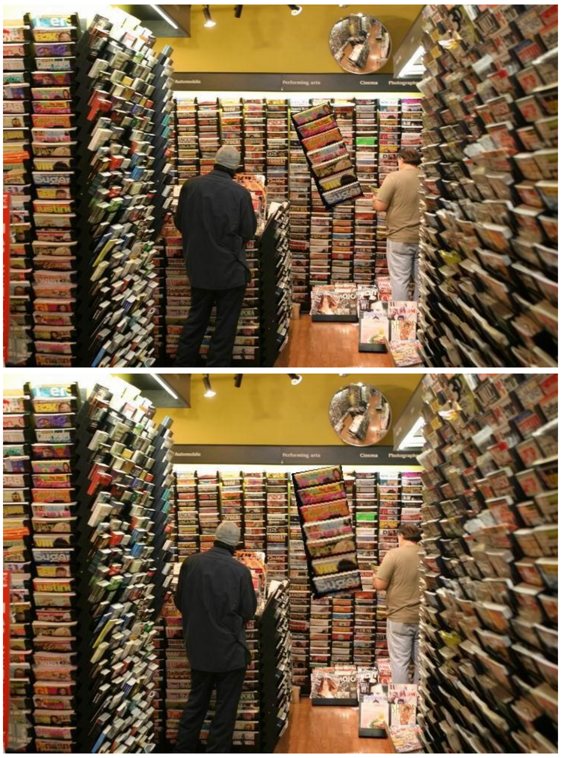


FIG 4.2. Tampered images (Tu)

Binary classes	Normal images	Tampered images
No.of images	110	110

Result

Figure 3 shows that count plot having similar classes of AU and TU

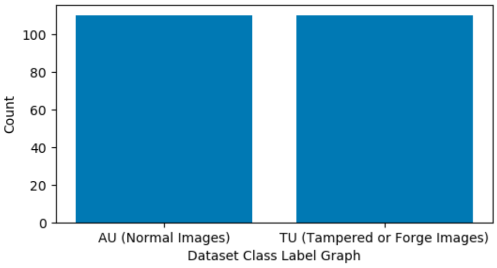


FIG 4.3. Count Plot

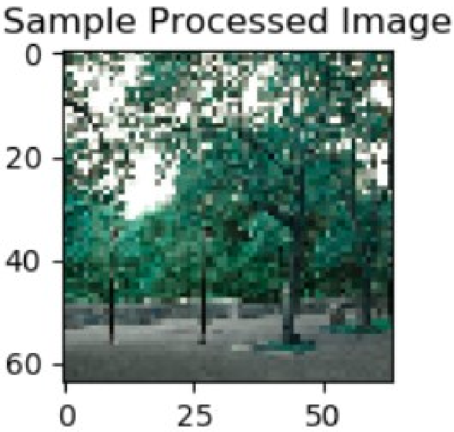
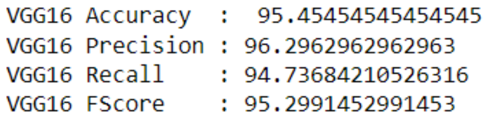


Figure 4.4: Sample Image

FIG 10.4 shows that sample image of Normal image



```
VGG16 Accuracy  :  95.45454545454545
VGG16 Precision : 96.2962962962963
VGG16 Recall    : 94.73684210526316
VGG16 FScore    : 95.2991452991453
```

Figure 4.5: Performance of CNN VGG16

Figure 10.5 shows that the image shows the performance of a CNN (Convolutional Neural Network) using the VGG16 algorithm. Here are the specific numbers:

- Accuracy: 95%
- Precision: 96.29%
- Recall: 95.45%
- F1 Score: 94.74%

These metrics are all very good, and they indicate that the CNN with VGG16 is performing well at classifying whatever data it was trained on.

- Accuracy: This is the overall percentage of correct predictions made by the model.
- Precision: This is the percentage of times that the model predicts a positive class and it is actually correct.
- Recall: This is the percentage of times that the model correctly identifies a positive class.
- F1 Score: This is a harmonic mean between precision and recall, and it is a way of combining these two metrics into a single score.

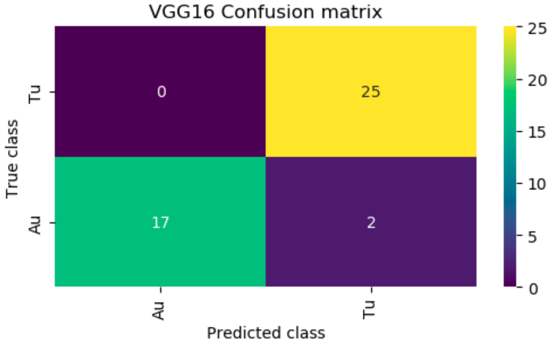


FIG 4.5: Confusion Matrix of CNN with CNN16

VGG16 is a pre-trained convolutional neural network model that can be used as a feature extractor for various computer vision tasks. It was developed by the Visual Geometry Group at Oxford University.

Here’s a breakdown of the values in the confusion matrix:

- **Predicted class** represents the classes the model predicted
- True class represents the ground truth which are the actual classes
- **Au** and **Tu** on the far right represent the different classes

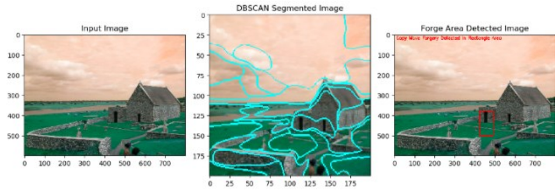


FIG 4.6: Output 1st

Figure 10. 6 and 10.7 shows the Forge area detected with CNN with VGG16 model and using with DBSCAN Segmented image

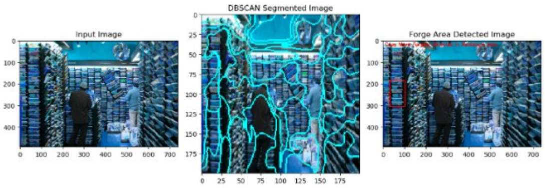


FIG 4.7: Output 2nd

V. CONCLUSION

The implementation of deep learning techniques for detecting copy-move forgery marks a significant leap in digital forensics. Utilizing VGG16, a convolutional neural network, allows for automated forgery detection with impressive accuracy. VGG16's hierarchical architecture effectively extracts both local and global features, enabling it to identify subtle image manipulations. This automation addresses the limitations of manual analysis, which is often time-consuming and prone to human error. By harnessing machine learning, we can efficiently process large datasets, maintaining the integrity of digital media in real-time applications, such as legal investigations and journalistic verifications.

Experimental results demonstrate the model's robustness. Trained on the MICC-F220 dataset, which includes a mix of authentic and tampered images, VGG16 achieved high accuracy, precision, recall, and F1 scores. Its proficiency in distinguishing between genuine and forged areas is evident. The integration of DBSCAN for clustering and key point extraction further enhances forgery detection. This combined approach of deep learning and advanced clustering techniques provides a thorough analysis of images, confidently identifying tampered areas.

The DBSCAN-based image segmentation, coupled with super pixel modification and pattern matching, allows for precise identification of forgery locations. This method visually marks forged areas, simplifying the understanding of the manipulation's extent and nature. Such visualizations reinforce the credibility of the automated system, making it a valuable asset for forensic experts and digital content verifiers.

In summary, the deep learning approach to copy-move forgery detection using VGG16 and DBSCAN offers an efficient solution to the challenges of manual analysis. With its high accuracy and reliability, it represents a crucial advancement in digital forensics. The capability to quickly and accurately analyze large volumes of images will significantly enhance forensic investigations, promoting the integrity of digital media

REFERENCES

[1]. A. Diwan and A. K. Roy, "CNN-Key point Based Two-Stage Hybrid Approach for Copy-Move Forgery Detection," in IEEE Access, vol. 12, pp. 43809-43826, 2024.

[2]. S. Booshehrian and E. Amiri, "Copy-Move forgery detection and classification using SRVM," 2024 10th International Conference on Artificial Intelligence and Robotics (QICAR), Qazvin, Iran, Islamic Republic of, 2024.

[3]. K. H. Hingrajiya and C. Patel, "An Approach for Copy-Move and Image Splicing Forgery Detection using Automated Deep Learning," 2023 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2023.

[4]. D. Narayan, Himanshu and R. Kamal, "Image Forgery Detection," 2023 International Conference on Disruptive Technologies (ICDT), Greater Noida, India, 2023.

[5]. M. Patel, K. Rane, N. Jain, P. Mhatre and S. Jaswal, "Image Forgery Detection using CNN," 2023 3rd International Conference on Intelligent Technologies (CONIT), Hubli, India, 2023.

[6]. S. Gazzah, L. R. Haddada, I. Shallal and N. E. B. Amara, "Digital Image Forgery Detection with Focus on a Copy-Move Forgery Detection: A Survey," 2023.

[7]. K. M. Hosny, A. M. Mortda, M. M. Fouda and N. A. Lashin, "An Efficient CNN Model to Detect Copy-Move Image Forgery," in *IEEE Access*