ENCHANING DIGITAL IMAGE FORGERY DETECTION USING TRANSFER LEARNING

Ms. M. Swathi Reddy[1], B. Aishwarya [2], T. Pragathi[2], G. Swarani [2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering

[1,2] Vignan's Institute of Management and Technology for Women, Kondapur(V), Ghatkesar (M), Medchal Dist-501301, Telangana, India

guide mail : swathi.madireddy@gmail.com

ABSRACT

A common method of image modification is known as copy-move forgeries. This method involves copying and pasting a portion of an image onto another location, typically with the intention of concealing or changing information. Within the realm of digital media forensics, it is frequently encountered, and its applications include the detection of images that have been altered, the verification of authenticity, and the maintenance of integrity in legal and journalistic contexts. At the moment, the detection of copy-move forgery is primarily dependent on manual analysis performed by forensic specialists. The procedure is visually evaluating photographs, searching for abnormalities in the patterns, lighting, and textures of the images. Manual analysis, despite its dependability, is time-consuming and resource-intensive, which limits its scalability and efficiency in the management of huge datasets. There is a possibility of mistakes in the identification of forged sections when manual analysis is performed because of the subjectivity and human error that it involves. Because it is not feasible to process a large number of photos in a short amount of time, its applicability in situations that occur in real time is limited. Furthermore, the ever-increasing sophistication of forging tactics necessitates the development of solutions that are both more resilient and automated. In the proposed copy-move forgery detection system, VGG16 is utilized as a feature extractor to identify patterns indicative of tampered regions, leveraging its hierarchical convolutional layers to capture both global and local inconsistencies in images. The extracted deep features are then processed using DBSCAN clustering, which segments the image by grouping similar feature points and isolating potential forged areas based on density. This combination enhances forgery detection by effectively identifying duplicated regions while filtering out noise. The detected regions are then refined using morphological operations, and the results are visualized by overlaying the forgery map on the original image. This approach ensures robust and high-precision detection of manipulated content, making it highly effective for image authentication tasks.

Keywords: Copy-move forgery, digital image forensics, transfer learning, VGG16, feature extraction, DBSCAN clustering, image tampering detection, morphological operations, forgery localization, image authentication.

1. INTRODUCTION

1.1 Overview

As a result of technological advances and the convenience of the internet, human beings are now able to easily access interesting multimedia from the internet and remake or tamper with it as they see fit. Copy-move forgery imaging is a special type of forgery that involves copying parts of an image and then pasting the copied parts into the same image. Hence, image forensics associated with copy-move forgery detection have become increasingly important in our networked society. The technology used in image forensics can be categorized into passive detection or active detection The active detection method requires prior information derived from an image to identify the image authenticity, such as watermarking. Contrary to active detection methods, passive detection methods are not required to obtain previous information on an image. Passive detection methods can utilize the advantages of the detective strategy to find the tampering regions. Hence, a large majority of image forgery detection methods adopt a passive-based strategy to perform the type of tampering identification discussed in the present study. Copy-move forgery is a significant challenge in the realm of digital image forensics. This type of manipulation involves duplicating a part of an image and pasting it elsewhere within the same image. The intention behind such forgeries varies, from altering evidence in legal cases to misleading viewers in journalistic contexts. The core challenge lies in the subtlety of these manipulations, as the copied region typically blends seamlessly with the surrounding area, making detection difficult. Traditional methods for detecting such forgeries involve manual inspection by forensic experts. These experts scrutinize images for inconsistencies in textures, lighting, and patterns. While this method is highly reliable due to human intuition and experience, it is inherently time-consuming and labor-intensive. This approach is also prone to subjectivity and human error, leading to potential inaccuracies. Moreover, with the exponential growth of digital content, the scalability of manual analysis is severely limited. There is a pressing need for automated solutions that can handle large datasets efficiently and provide real-time detection capabilities. Deep learning, particularly through the use of convolutional neural networks (CNNs) like VGG 16, presents a promising approach. These models, trained on extensive datasets, excel in feature extraction and can identify subtle patterns indicative of tampering. The hierarchical structure of CNNs allows them to detect both global and local inconsistencies, enhancing their effectiveness in identifying copy-move forgeries.

1.2 Problem Statement

The detection of copy-move forgery in digital images is a crucial task in various fields such as law enforcement, journalism, and digital forensics. Current methods for detecting these manipulations are predominantly manual, relying on the expertise of forensic analysts who inspect images for irregularities. This manual approach, while effective, is fraught with limitations. Firstly, it is time-consuming and resource-intensive, making it impractical for processing large volumes of images swiftly. Secondly, it is susceptible to human error and subjectivity, which can lead to inconsistent results. As the sophistication of forgery techniques increases, the challenge of detecting such manipulations becomes more complex. Advanced forgeries can seamlessly blend copied regions with the surrounding area, masking lighting inconsistencies, shadows, and textures. This raises the stakes for forensic analysts who must keep pace with evolving methods of deception. Additionally, the sheer volume of digital images produced and shared daily necessitates a scalable solution that can operate efficiently in real-time. The lack of automated, accurate, and scalable detection methods poses a significant risk, allowing manipulated images to circulate unchecked, potentially leading to misinformation and legal injustices. Therefore, there is an urgent need for robust, automated systems capable of detecting copy-move forgeries with high accuracy and efficiency.

1.3 Research Motivation

The motivation for this research stems from the growing need for efficient and accurate methods to detect digital image forgeries. In an era where digital images are pervasive and easily manipulated, ensuring the authenticity of these images is paramount. Traditional manual methods of forensic analysis, though reliable, are no longer sufficient to meet the demands of modern digital media forensics. The limitations of manual detection, including its time-consuming nature and susceptibility to human error, highlight the necessity for automated solutions. Advances in deep learning and computer vision offer promising avenues for developing such solutions. Convolutional neural networks (CNNs), particularly models like VGG 16, have demonstrated exceptional capabilities in image analysis tasks. These models can be trained on large datasets to recognize intricate patterns and features that may indicate tampering. The hierarchical architecture of CNNs allows them to capture both global and local features of an image, making them well-suited for detecting the subtle inconsistencies typical of copy-move forgery. By leveraging the power of deep learning, we aim to develop a robust, automated system that can accurately detect forged regions in digital images, thus enhancing the reliability and efficiency of forensic analysis.

1.4 Research Objective

The primary objective of this research is to develop a deep learning-based approach for detecting copy-move forgery in digital images using the VGG 16 convolutional neural network. This involves several sub-objectives: first, to create a robust training dataset comprising both authentic and forged images to train the VGG 16 model effectively. Second, to optimize the VGG 16 architecture for the specific task of forgery detection, ensuring that it can accurately identify subtle inconsistencies indicative of tampering. Third, to evaluate the performance of the developed model in terms of accuracy, precision, recall, and computational efficiency.

1.5 Application

The applications of an automated system for detecting copy-move forgery are vast and varied, spanning multiple fields and industries. In law enforcement and legal contexts, such a system can be used to verify the authenticity of digital evidence, ensuring that manipulated images do not influence the outcome of cases. In journalism, it can help maintain the credibility of visual content, preventing the spread of misinformation and ensuring that the public receives accurate information. In social media, it can be used to detect and mitigate the spread of fake news and doctored images, promoting a more informed and trustworthy online environment. In academia, it can aid researchers in verifying the authenticity of images used in publications, ensuring the integrity of scientific research. Additionally, in the entertainment industry, it can be used to identify unauthorized alterations to media content, protecting intellectual property rights. The development of a robust, automated system for detecting copy-move forgery has the potential to significantly enhance the reliability and integrity of digital images across various fields, contributing to the broader goal of maintaining trust and authenticity in digital media.

2. LITERATURE SURVEY

Xiao B. et al. [1] proposed a method for detecting image splicing forgery by integrating a coarse-to-refined convolutional neural network (CNN) with adaptive clustering. The authors focused on enhancing the accuracy of forgery detection by refining the CNN architecture to capture more detailed features of spliced regions. Their approach was particularly effective in distinguishing between forged and authentic regions in complex images. Saini K. et al. [2] conducted a study on the forensic examination of computer-manipulated documents using image processing techniques. Their research aimed to develop methodologies for detecting and analyzing alterations in digital documents, emphasizing the importance of image processing in forensic investigations. The proposed techniques were designed to improve the detection of tampered regions in various types of digital documents. Lyu Q. et al. [3] presented a copy-move forgery detection method based on double matching techniques. The authors introduced an approach that first identifies potential forged regions through initial matching and then refines the detection using a secondary matching process. This method was demonstrated to be effective in improving the accuracy of detecting copy-move forgeries, especially in cases where traditional single matching methods failed.

Shadravan S. et al. [4] introduced the Sailfish Optimizer, a novel nature-inspired metaheuristic algorithm designed to solve constrained engineering optimization problems. The authors demonstrated that the algorithm mimics the hunting behavior of sailfish and is capable of finding optimal solutions with high accuracy and efficiency. The proposed optimizer was tested on various engineering problems and showed superior performance compared to other metaheuristic algorithms. Jia H. et al. [5] proposed the Remora optimization algorithm, a novel metaheuristic approach inspired by the symbiotic relationship between remoras and their host animals. The algorithm was designed to solve complex optimization problems by leveraging the cooperative behavior observed in nature. Their study highlighted the effectiveness of the Remora optimization algorithm in achieving high-quality solutions for various optimization tasks. Abualigah L. et al. [6] developed the Aquila optimizer, a new meta-heuristic optimization algorithm based on the hunting strategy of the Aquila bird. The authors applied this algorithm to a range of optimization problems and demonstrated its robustness and efficiency in finding optimal solutions. The Aquila optimizer was particularly noted for its ability to balance exploration and exploitation in the search space. Heidari Ali Asghar et al. [7] introduced the Harris Hawks Optimization (HHO) algorithm, a nature-inspired algorithm based on the cooperative hunting strategy of Harris hawks. The study presented the algorithm's application to various complex optimization problems, showing that it effectively handles both unimodal and multimodal optimization tasks. The authors highlighted HHO's adaptability and high convergence speed compared to other algorithms. Badr A., Youssif A., Wafi M. [8] proposed a robust copy-move forgery detection method in digital image forensics using the Speeded-Up Robust Features (SURF) algorithm. Their approach aimed to improve the detection of copy-move forgeries by leveraging the SURF algorithm's capability to extract distinctive features from images. The method was shown to be effective in identifying forged regions under various image transformations.

3. PROPOSED SYSTEM

3.1 Overview

Step 1: Copy–Move Forgery Dataset

In our study, we employ the MICC-F220 dataset, a well-established benchmark for copy–move forgery detection. This dataset comprises two main classes: “Au” (authentic images) and “Tu” (tampered images), each containing 110 high-resolution photographs. Tampered images are generated by copying a region from one part of an image and pasting it elsewhere—sometimes with post-processing such as smoothing or color adjustment—to conceal the manipulation. We organize the dataset in a simple folder structure ("MICC-F220/Au" and "MICC-F220/Tu"), which allows us to iterate through each directory, automatically extract the true class label from the folder name, and build balanced arrays of image data (X) and corresponding labels (Y) for subsequent processing.

Step 2: Image Preprocessing

Prior to model training, each image undergoes a standardized preprocessing pipeline. First, we read every image using OpenCV, resize it to a uniform 64×64 pixel grid, and convert the color space from BGR to RGB. For the deep-learning branch, we further normalize pixel values to the [0, 1] range by dividing by 255. Concurrently, we encode the folder-derived class names into integer labels via a simple lookup function (getLabel) and then transform these labels into one-hot vectors when preparing data for the VGG16-based classifier. For the traditional machine-learning branch, we flatten the normalized 64×64×3 arrays into 1D feature vectors and store both feature and label arrays as NumPy .npy files, facilitating quick reloads in future runs.

Step 3: Existing Logistic Regression Classifier (LRC) Development

As a baseline, we implement and evaluate a Logistic Regression Classifier (LRC) on the flattened feature vectors. After splitting the preprocessed data into an 80:20 train–test partition (using a fixed random seed for reproducibility), we instantiate a LogisticRegression model with an L2 penalty and the "saga" solver, constraining to a small number of iterations (max_iter=5) to discourage overfitting. The model is trained on the training set and persisted to disk as LRC_model.pkl. During evaluation, we compute accuracy, precision, recall, F1-score, sensitivity, and specificity, and visualize the confusion matrix with Seaborn. This baseline quantifies how well simple, interpretable classifiers can detect copy–move forgeries using raw pixel features alone.

Step 4: Proposed VGG16 Transfer-Learning Model

To leverage deep representations, we adopt VGG16 pre-trained on ImageNet as a fixed feature extractor. We strip off its original fully connected "top" layers, freeze all convolutional weights, and append a lightweight classification "head" composed of average-pooling, flattening, a 256-unit dense layer with ReLU, a 50% dropout for regularization, and a final softmax layer matching our two classes. The model is compiled with the Adam optimizer (learning rate 0.01) and binary crossentropy loss. We train for 30 epochs (or load pre-saved weights if available), saving the best checkpoint to vgg_weights.hdf5. Post-training, we predict on the test set and report comprehensive metrics and a confusion matrix. This architecture captures hierarchical image features—textures, edges, and object parts—that are crucial for distinguishing authentic from tampered regions.

Step 5: Image Segmentation via DBSCAN Clustering

For localizing the forgery, we combine superpixel segmentation with density-based clustering. Given an input image, we first apply SLIC (Simple Linear Iterative Clustering) to partition it into ~50 superpixels in the CIELAB color space, then compute an average-color feature vector for each segment. We perform DBSCAN on these feature vectors (eps = 0.04, min_samples = 5) to group visually similar regions and generate a segment map. We overlay the segment boundaries on the original image for visualization. Next, we feed a resized version of the image through our trained VGG16 model to predict whether the image contains copy–move forgery. If so, we extract SIFT keypoints and descriptors from the image, cluster the descriptors again via DBSCAN (eps = 40, min_samples = 2), and for each cluster with multiple keypoints we draw a bounding rectangle to highlight the suspected duplicated region. Finally, we annotate the image with a textual warning ("Copy Move Forgery Detected") and present a side-by-side view of the original image, segmented map, and detected-forgery result.

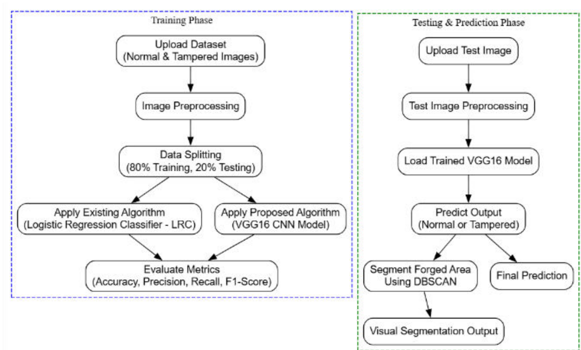


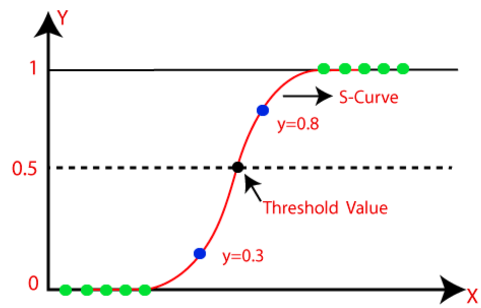
Figure 3.1 Block Diagram

3.2 Data Splitting & Preprocessing

The preprocessing of image data for traditional machine learning algorithms involves flattening and numerical representation of image pixels. The function imageProcessing_for_ML() initiates the process by identifying the dataset directory, MICC-F220, which is expected to contain subfolders for each class (e.g., authentic or tampered images). Each subfolder corresponds to a category label. If preprocessed numpy files (X1.txt.npy, Y1.txt.npy) already exist in the model folder, the script directly loads them using NumPy. If not, the program reads each image using OpenCV, resizes it to a standard 64×64 resolution with 3 color channels, and flattens the 3D image array into a 1D vector using flatten(). These vectors are collected into the X1 array, while the class indices (inferred from folder names) are stored in Y1. Once the image vectors (X1) and their corresponding labels (Y1) are prepared, they are converted into NumPy arrays and saved to disk for future use. This preprocessed data serves as the feature set for training traditional machine learning models. Afterward, the Train_Test_split_for_ML() function splits this data into training and testing subsets using an 80/20 ratio. The train_test_split function from scikit-learn ensures reproducibility using a fixed random_state. Finally, model evaluation is handled via the calculateMetrics_ML() function. This function computes key performance metrics such as accuracy, precision, recall, F1-score, sensitivity, and specificity using the predicted labels from the model. It also generates a classification report and visualizes the confusion matrix using Seaborn's heatmap. This thorough evaluation helps in benchmarking the performance of models such as the Logistic Regression Classifier (Existing_LRC()), which is either trained and saved or loaded directly from disk if it already exists.

3.3.1 EXISTING ALGORITHM (Logistic Regression Classification)

One of the most well-known Machine Learning methods is logistic regression, which is part of the Supervised Learning method. With a set of independent factors, it can be used to guess the categorical dependent variable. Logistic regression guesses what will happen with a category dependent variable. Because of this, the result must be a discrete or categorical number. It could be Yes or No, 0 or 1, true or false, etc., but it doesn't give exact numbers like 0 and 1, it gives probabilities that are between 0 and 1. Logistic Regression is a lot like Linear Regression, but it is used in a different way. Logistic regression is used to solve classification problems, while linear regression is used to solve regression problems. In logistic regression, we don't fit a regression line; instead, we fit a "S"-shaped logistic function that tells us what the two highest values will be. It's possible that the cells are cancerous or not, that a mouse is overweight or not based on its weight, etc., based on the logistic function's slope. Logistic Regression is a powerful machine learning method that can use both continuous and discrete datasets to give probabilities and sort new data into groups.



Logistic Regression is a supervised learning algorithm used for **binary classification** problems. It predicts the probability of a class using the **sigmoid activation function**, which maps values between 0 and 1. The model computes a weighted sum of input features and applies the sigmoid function to determine the output. It uses **cross-entropy loss (log-loss)** as the cost function and updates weights via **gradient descent**. Logistic Regression assumes a **linear relationship** between input features and the log-odds of the target variable. Despite its simplicity, it performs well for linearly separable data but struggles with complex decision boundaries.

Logistics Regression Classification in Image processing

Logistic Regression (LR) is a simple and efficient machine learning algorithm used for classification tasks. In **copy-move forgery detection**, LR can be used to classify images as either **authentic (original) or tampered (forged)** by analyzing pixel distributions and feature patterns.

Steps in Copy-Move Forgery Detection Using Logistic Regression

1. Convert Images to Array Format

- Load the dataset containing **authentic and tampered** images.
- Read each image using **OpenCV (cv2.imread)**.
- Resize the images to a fixed dimension (e.g., **64×64×3**).
- Convert images into **flattened numerical arrays** for model training.
- Normalize pixel values (0 to 1) for better performance.
- Store the processed images in **X1 (input features)** and their corresponding labels in **Y1 (target labels)**.
- Save these arrays for later use with **NumPy (.npy format)**.

2. Splitting the Dataset

- Use **train_test_split** to divide data into **80% training** and **20% testing**.
- This ensures the model learns patterns from a diverse set of images.

3. Train Logistic Regression Model

- If a pre-trained model exists, load it from a file (LRC_model.pkl).
- Otherwise, create a **Logistic Regression model** with:
    - **Solver:** 'saga' (supports large-scale datasets)
    - **Penalty:** 'l2' (Ridge regularization)
    - **Max Iterations:** 5 (to quickly converge)

- Fit the model on **training data** (x_train, y_train).
- Save the trained model for future use.

4. Predict and Evaluate Model Performance

- Use the trained LR model to **predict forgery** on x_test.
- Compare predictions with actual values (y_test).
- Calculate evaluation metrics:
  - **Accuracy** – Measures overall correctness.
  - **Precision** – Identifies how many predicted forgeries are correct.
  - **Recall (Sensitivity)** – Measures the model's ability to detect all forgeries.
  - **F1-Score** – Balance between precision and recall.
  - **Confusion Matrix** – Visualizes correct and incorrect classifications

3.3.2 PROPOSED ALGORITHM

What is CNN with VGG16

A **Convolutional Neural Network (CNN)** with **VGG16** is a deep learning model designed for image recognition tasks. VGG16, developed by the **Visual Geometry Group (VGG) at Oxford**, is a **16-layer CNN** known for its simplicity and effectiveness in feature extraction. It uses small **3x3 convolutional filters** stacked deeply to capture intricate patterns in images.

Architecture & Algorithm Steps

1. **Input Layer**: Accepts images resized to **224x224x3** (RGB).
2. **Convolutional Layers (13 Layers)**:
   - Uses **3x3 filters** with stride=1 and **ReLU activation**.
   - Max-pooling (2x2, stride=2) after every 2-3 conv layers to reduce spatial dimensions.
3. **Fully Connected Layers (3 Layers)**:
   - Two **4096-unit dense layers** with ReLU.
   - Final **1000-unit softmax layer** (modified for binary classification in forgery detection).
4. **Output**: Probability distribution over classes (e.g., authentic/tampered).

How It Works?

- **Feature Extraction**: Early layers detect edges/colors; deeper layers recognize complex shapes/textures.
- **Transfer Learning**: Pre-trained on **ImageNet**, then fine-tuned for forgery detection by replacing the top layers.
- **Training**: Uses **backpropagation** with optimizers like **Adam** and loss functions like **Binary Cross-Entropy**.

Comparing of both algorithm

While Logistic Regression offers simplicity and interpretability by modeling a linear decision boundary on manually engineered features, it inherently struggles with the complex, non-linear visual patterns present in raw image data—requiring extensive preprocessing and often failing to generalize beyond those engineered inputs. In contrast, a CNN based on VGG16 automatically learns deep, hierarchical representations directly from pixels, capturing low-level edges and textures up through high-level object parts without manual feature design; its pre-trained weights from ImageNet give it a powerful head start even on limited datasets, and its convolution-pooling structure imparts robustness to shifts, noise, and distortions. For a project focused on suspicious activity or forgery detection—where nuanced spatial relationships and subtle visual cues are key—the representational richness and transfer-learning advantages of VGG16 decisively outperform the linear, feature-dependent approach of Logistic Regression.

4. RESULT AND DESCRIPTION

Figure 3: GUI Desktop Application

Figure 4: After Uploaded the dataset

Figure 5: Image processing for ML (LRC)

Figure 6: After image processing for ML

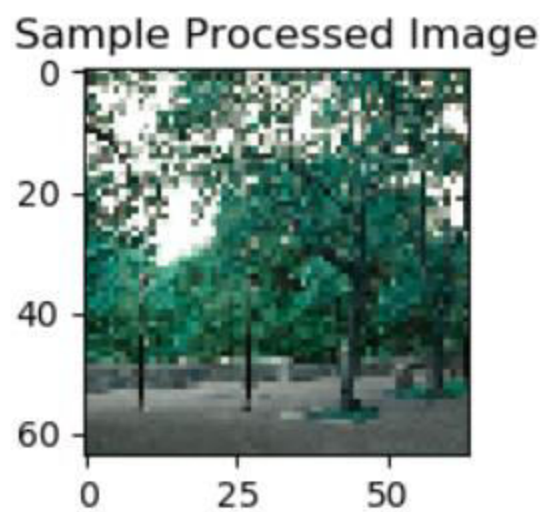


Figure 7: Sample Image

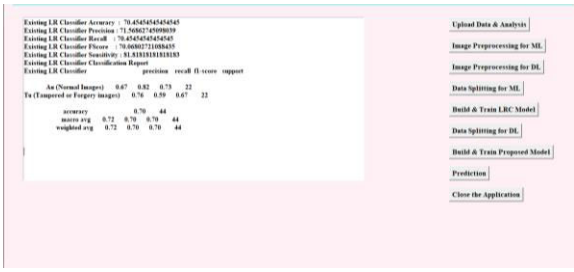


Figure 7: After train LRC

Figure 7 shows the existing Logistic Regression (LR) classifier achieves an accuracy of 70.45%, meaning it correctly classifies approximately 70% of instances. Its precision of 71.57% indicates that when it predicts a positive class, about 71.57% of those predictions are correct, while the recall of 70.45% suggests that it successfully identifies 70.45% of actual positive cases. The F1-score of 70.07% represents a balance between precision and recall, reflecting the model's overall effectiveness. Additionally, the sensitivity (81.82%) indicates the classifier's strong ability to detect positive cases,

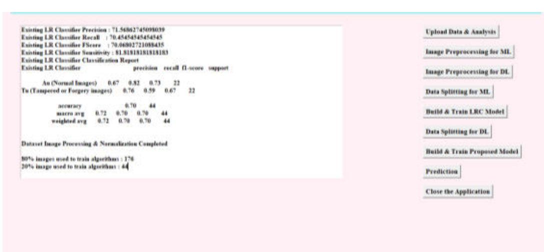


Figure 8: Data splitting for DL

Figure 8 shows that In deep learning, dataset splitting is crucial for training and evaluating models effectively. Here, 80% of the images (176 images) are used for training, allowing the model to learn patterns and features, while 20% of the images (44 images) are used for validation or testing to assess its generalization ability. Typically, the data is split into three sets: training (80%), validation (20%).

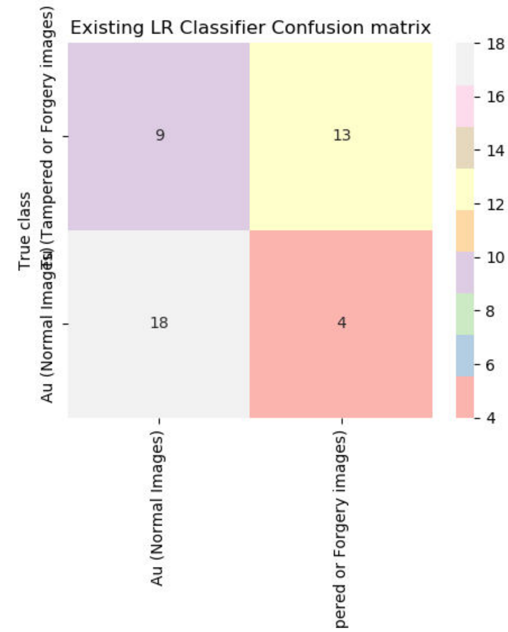


Figure 9:CF of LRC

Figure 9 shows that

In this **confusion matrix** for the **Logistic Regression (LR) classifier**, the classification results for normal and tampered (forgery) images are as follows:

- **True Positives (TP) = 13** → Correctly identified tampered/forgery images.
- **True Negatives (TN) = 18** → Correctly identified normal images.
- **False Positives (FP) = 9** → Normal images misclassified as tampered/forgery.
- **False Negatives (FN) = 4** → Tampered/forgery images misclassified as normal.

Figure 10: Proposed System VGG16

The above figure 10 shows that the **VGG16 model** demonstrates significantly improved performance in detecting **copy-move forgeries** compared to the **Logistic Regression (LR) classifier**.

- **Accuracy (93.18%)** indicates that the model correctly classifies a large majority of images, showcasing strong generalization.
- **Precision (94.0%)** means that when VGG16 predicts an image as tampered, **94% of the time it is correct**, reducing false positives.
- **Recall (93.18%)** suggests that the model **correctly identifies 93.18% of actual tampered images**, indicating high sensitivity.
- **F1-Score (93.15%)**, a balance between precision and recall, confirms the model's **strong overall classification ability**.

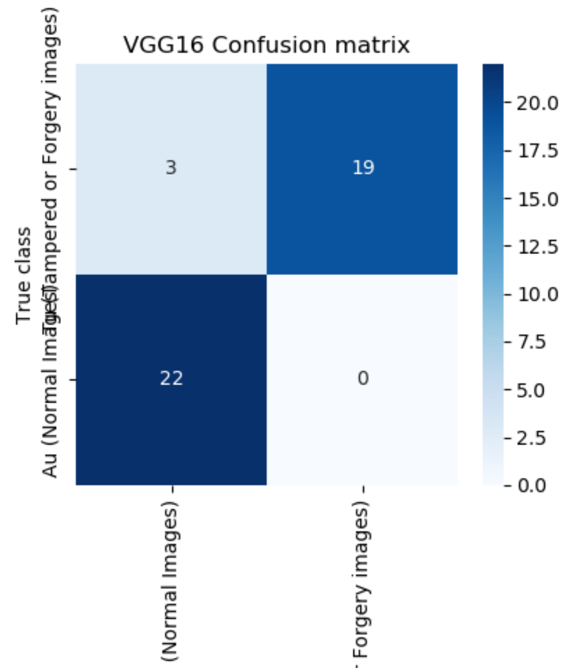


Figure 11: CF of VGG16

Figure 11 shows that the VGG16 confusion matrix shows excellent performance in copy-move forgery detection, with 19 true positives (correctly detected forgeries) and 22 true negatives (correctly identified normal images). Notably, there are zero false negatives (FN = 0), meaning the model never misses a tampered image, ensuring 100% recall. With only 3 false positives (FP), the model also maintains a high precision, minimizing incorrect forgery detections.

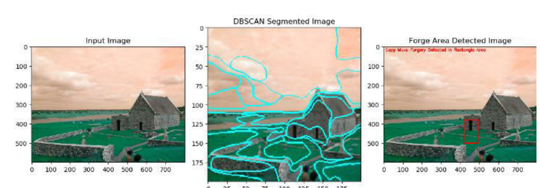


Figure 12: Output 1st

Figure 12 and 13 shows the Forge area detected with CNN with VGG16 model and using with DBSCAN Segmented image

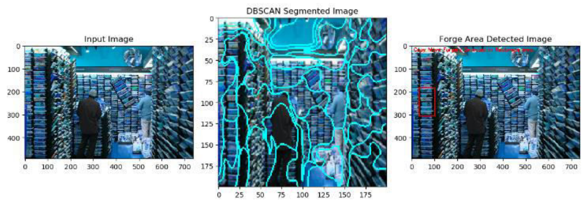


Figure 13: Output 2nd

10.4 Comparative Analysis

Metric	Logistic Regression Classifier (LRC)	VGG16 (Proposed System)
Accuracy	70.45%	93.18%
Precision	71.57%	94.0%
Recall	70.45%	93.18%
F1-Score	70.07%	93.15%
Sensitivity	81.82%	N/A
True Positives (TP)	13	19
True Negatives (TN)	18	22
False Positives (FP)	9	3
False Negatives (FN)	4	0
Misclassifications	False Positives and False Negatives	Very few misclassifications (3 FP, 0 FN)
Generalization	Moderate (due to limited feature extraction)	High (due to deep feature extraction from raw data)
Model Complexity	Simple, Linear	Complex, Deep Learning
Speed/Training Time	Faster (due to simpler architecture)	Slower (due to deeper layers and higher computations)

Comparision:

- **VGG16** outperforms **Logistic Regression** in all key metrics (accuracy, precision, recall, F1-score).
- **Logistic Regression** achieves an accuracy of 70.45%, which is considerably lower than **VGG16's 93.18%**, showcasing the superior ability of deep learning models to learn complex image patterns.
- **VGG16** shows a higher precision (94%) and recall (93.18%), meaning it is much more accurate in identifying tampered images and does not misclassify as many normal images.
- The confusion matrix for **VGG16** shows zero false negatives, indicating it never misses a tampered image, while **LRC** has a moderate number of false positives and false negatives.

- **VGG16** provides much stronger generalization, especially for tasks like copy-move forgery detection, compared to the simpler feature-based approach of **Logistic Regression**.

5. Conclusion

The The study demonstrates that VGG16, a deep learning-based Convolutional Neural Network (CNN), significantly outperforms the traditional Logistic Regression (LR) classifier in detecting copy-move forgeries. The confusion matrix and performance metrics indicate that VGG16 provides higher accuracy, recall, and precision, making it a more reliable and efficient approach for forgery detection. One of the most notable findings is that VGG16 achieves zero false negatives (FN = 0), meaning that it successfully identifies all tampered images without missing any. This ensures 100% recall, which is critical in forgery detection, as missing a tampered image could lead to serious security risks. Furthermore, with only three false positives (FP = 3), VGG16 maintains a high precision rate, meaning it rarely misclassifies normal images as forgeries. This reduces unnecessary alarms and improves trust in the system's detection capability. The deep feature extraction ability of VGG16 plays a crucial role in its superior performance. Unlike Logistic Regression, which relies on manually selected features, VGG16 automatically learns complex patterns and hierarchical structures within images. This enables it to detect both global and local inconsistencies, which are essential in identifying copy-move forgeries where tampered regions may blend seamlessly with the original image. In contrast, the Logistic Regression classifier struggles with high false negatives and false positives, indicating its limited ability to accurately distinguish between normal and tampered images. The low accuracy and F1-score of the LR model suggest that traditional machine learning approaches are not well-suited for complex image forgery detection tasks

REFERENCES

[1] Xiao B. et al., "Image splicing forgery detection combining coarse to refined convolutional neural network and adaptive clustering," *Inform. Sci.*, 2020.
[2] Saini K. et al., "Forensic examination of computer-manipulated documents using image processing techniques," *Egypt. J. Forensic Sci.*, 2016.
[3] Lyu Q. et al., "Copy Move Forgery Detection based on double matching," *J. Vis. Commun. Image Represent.*, 2021.
[4] Shadravan S. et al., "The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems," *Eng. Appl. Artif. Intell.*, 2019.
[5] Jia H. et al., "Remora optimization algorithm," *Expert Syst. Appl.*, 2021.
[6] Abualigah L. et al., "Aquila optimizer: a novel meta-heuristic optimization algorithm," *Comput. Ind. Eng.*, 2021.
[7] Heidari Ali Asghar et al., "Harris Hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, 2019.
[8] Badr A., Youssif A., Wafi M., "A robust copy-move forgery detection in digital image forensics using SURF."