Detection of image manipulation using Feature-Based Algorithms

Ramasahayam Riddhima Reddy
Department of electronics and communications engineering
MGIT college
Hyderabad, India
email:riddhimareddy1@gmail.com

***Abstract*—** In today's digital age, the integrity of images is paramount as they serve various critical purposes, from news reporting to personal communication and marketing. With the widespread availability of powerful image editing tools, the risk of image manipulation, particularly copy and move forgery, has increased. This deceitful practice involves copying or moving objects within images to create manipulated versions, potentially deceiving viewers by altering the context or composition. To address this challenge, this research project is dedicated to image forgery detection with a specific focus on identifying copy and move forgery. We propose a comprehensive solution that leverages digital image processing techniques, encompassing image preprocessing, feature point extraction, feature matching, and the clustering of similar features. Our approach incorporates advanced algorithms, such as the Stationary Wavelet Transform (SWT) for image preprocessing, the Scale-Invariant Feature Transform (SIFT) for feature point extraction, and the Geometric Two Nearest Neighbour (G2NN) for feature matching. Additionally, we employ hierarchical clustering to create clusters of similar feature points, enabling us to pinpoint potential forgery regions. In summary, the aim to provide a robust and effective method for detecting and mitigating image manipulation, with a particular focus on copy and move forgery. By achieving accurate detection of such manipulations, we contribute to preserving the authenticity and integrity of digital images, thereby protecting against misleading and deceptive uses of altered images.

Keywords*-digital image processing, image forgery, SIFT, G2NN*

I. INTRODUCTION

In the contemporary digital landscape, the veracity of images stands as an issue of paramount significance. Image forgery detection, the discipline dedicated to discerning any form of image manipulation, plays a pivotal role in safeguarding the integrity of visual content. This is especially vital in a world where digital images serve multifaceted purposes, spanning from the dissemination of news, personal communication, to marketing endeavours. The proliferation of powerful image editing tools has ushered in an era where image alterations can be effortlessly executed, giving rise to a landscape replete with the potential for misinformation and deception.

The imperative of image forgery detection is underscored by its far-reaching consequences. In the realm of journalism, the authenticity of visual content is the linchpin upon which the public's trust in the media is built. In the corporate sphere, deceptive or spurious images can precipitate financial losses and inflict damage upon a company's reputation. On a personal level, the implications of image forgery are profound, extending to wrongful accusations and irreparable harm to an individual's character.

Specifically, within the realm of image manipulation, copy and move forgery stands as a discerning challenge. This technique involves the meticulous act of copying and relocating objects within an image, thereby engendering a distorted iteration of the original visual composition. Such manipulations are often orchestrated to effectuate the removal or addition of objects or individuals within the frame, masterfully reshaping the image's context and narrative, all with the intent of beguiling and misleading the observer.

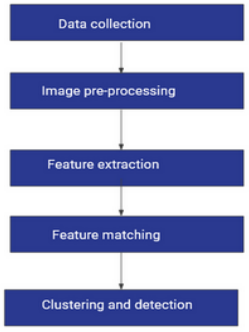
II. BACKGROUND

Image forgery detection has gained significant attention in recent years due to the widespread use of digital images in various domains. Copy and move forgery, a specific form of image manipulation, refers to the process of copying and pasting an object from one location in an image to another, or moving an object within the same image, in order to create a manipulated version of the original image. This type of forgery is often used to remove or add objects or people from an image, or to alter the context or composition of the image in a way that misleads or deceives the viewer.

Some of the traditional methods used were :

a. **Histogram Analysis**: Histogram analysis is a widely used technique in image forensics, primarily employed to detect global changes in an image's intensity distribution. It relies on the concept of the histogram, which represents the frequency of pixel intensity values in the image. The first step is to generate a histogram for the image. This histogram is a graph that shows the frequency of pixel intensity values along the x-axis and their corresponding frequencies (number of pixels) on the y-axis.

Histogram analysis is particularly effective in detecting global changes in pixel intensities. It can reveal shifts or differences in the histograms. For instance, if an image has been manipulated by altering contrast or brightness, this often results in a noticeable shift in the histogram. You can apply a threshold to determine whether the differences are significant enough to flag the image as potentially tampered. The choice of threshold depends on the specific application and the level of sensitivity required.



b. **Noise Analysis**: Image noise refers to random variations in pixel values that are not part of the intended image content. Forgers may inadvertently introduce new noise when tampering with an image. By comparing noise patterns, inconsistencies can be detected. Start by extracting the noise component of the image. This can be done by subtracting an estimate of the original, noise-free image from the suspicious image. Image noise refers to random variations in pixel values that are not part of the intended image content. Forgers may inadvertently introduce new noise when tampering with an image. By comparing noise patterns, inconsistencies can be detected. Start by extracting the noise component of the image. This can be done by subtracting an estimate of the original, noise-free image from the suspicious image. Apply a threshold to determine whether the differences in noise characteristics are significant enough to suggest tampering. The choice of threshold may depend on the specific application and the expected noise variations.

It's essential to understand that while these traditional methods are valuable for detecting simple forgeries, they often struggle with more sophisticated manipulations, hence increasing the need for more morphological processing.

III. METHODOLOGY .

In this paper, copy and move image forgery is detected using Stationary Wavelet Transform (SWT) for image preprocessing, the Scale-Invariant Feature Transform (SIFT) for feature point extraction, and the Geometric Two Nearest Neighbour (G2NN) for feature matching. Additionally, we employ hierarchical clustering to create clusters of similar feature points, enabling us to pinpoint potential forgery regions

a. data collection.

Gather a diverse dataset of digital images, including authentic images and images with known copy and move forgeries. These images should cover a range of scenarios and image types to ensure a comprehensive evaluation

sample image

b. image pre-processing

Pre-processing in image processing refers to the steps that are taken before the actual analysis or processing of an image. These steps are usually performed to improve the quality of the image, to enhance its features, or to make it more suitable for a specific task. In the preprocessing phase, the input image was initially imported and transformed into a NumPy array via the Python Imaging Library (PIL). The image was then dissected into its red, green, and blue colour channels, making it amenable to

individual processing. The Stationary Wavelet Transform (SWT) was subsequently applied to each colour channel, decomposing the image into four sub bands: LL, representing low-frequency components, and LH, HL, and HH, representing high-frequency components.

Following the SWT, the "LL," "LH," "HL," and "HH" sub band images were reconstructed from the respective colour channels, enabling the synthesis of complete RGB images. To ensure data consistency and compatibility, the "LL" image underwent normalization and conversion. The 'clip' function was employed to constrain pixel values within the 0 to 1 range, and the 'convert array' function converted the image into an 8-bit unsigned integer array. This preprocessing pipeline readied the image data for subsequent analysis, laying a solid foundation for the image forgery detection process by ensuring uniformity and facilitating further manipulations.

Stationary wavelet transform: It is a versatile mathematical technique used for signal and image processing. It provides a multi-resolution analysis of data, allowing for the exploration of information at different scales while preserving the original data size at each level of analysis.

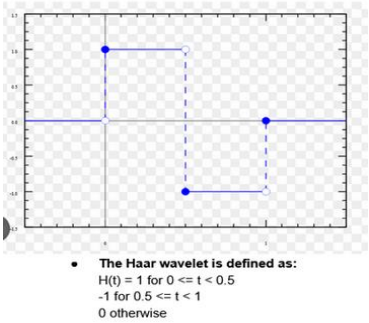
The key to understanding how the SWT works lies in its wavelet basis functions. These wavelets are mathematical functions with the ability to capture specific frequency components within data. By convolving the input signal or image with these wavelet functions, the SWT effectively decomposes the data into different frequency sub bands. The process starts with the choice of a wavelet function, and this choice depends on the application and the

nature of the data being analysed. Different wavelets are optimized for capturing various features and details within the data.

The decomposition yields sub bands, each containing unique frequency information. These sub bands are categorized as follows:

- **LL (Low-Low)**: The LL subband represents low-frequency components and provides an approximation of the overall trend in the data at the current scale.
- **LH (Low-High)**: In the LH subband, low-frequency components are found in the horizontal direction, while high-frequency components are present in the vertical direction.
- **HL (High-Low)**: The HL subband contains low-frequency components in the vertical direction and high-frequency components in the horizontal direction.
- **HH (High-High)**: The HH subband captures high-frequency details in both the horizontal and vertical directions, making it well-suited for preserving fine details and textures.

Once the subbands are obtained, they are stored in a data structure for further analysis. Researchers and practitioners can then apply various operations to these subbands, such as filtering, thresholding, or feature extraction, depending on the specific application.



Haar wavelet

Image after preprocessing

c. Feature extraction

Feature extraction is an important step in the process of detecting copies and moves in images, as it allows analysts and algorithms to identify and extract relevant and informative characteristics or features from the images that can be used to differentiate between copies and original images.

SIFT(Scale Invariant Feature Transform): The Scale-Invariant Feature Transform (SIFT) is a robust and widely used algorithm in computer vision for detecting and describing local features in images. It is particularly effective in scenarios where the scale and orientation of the features can vary significantly.

- **Scale-space extrema detection:** The SIFT algorithm begins by constructing a scale-space representation of the input image, which is a series of increasingly smoothed and down sampled versions of the image. At each scale, the algorithm looks for local extrema in the Difference of Gaussians (DoG)

  DoG(x, y, σ) = G(x, y, kσ) - G(x, y, σ)

  where G(x, y, σ) is the Gaussian function defined as:

  G(x, y, σ) = (1/(2πσ^2)) * exp(-(x^2 + y^2)/(2σ^2))

  The DoG function is used to highlight differences in the image at different scales. The local extrema of the DoG function correspond to potential interest points in the image.

- **Key point localization:** The locations of the extrema are refined to more accurately locate the key points in the original image. This is done by fitting a detailed model of the extrema to the nearby image data using a least squares optimization method.

  D(x, y, σ) = [Ixx(x, y, σ) Ixy(x, y, σ)] [Ixy(x, y, σ) Iyy(x, y, σ)]

  and (x, y, σ) is the vector of partial derivatives of the DoG function with respect to x, y, and σ. The Hessian matrix is used to model the curvature of the DoG function at the extrema.

  The key point locations are refined by minimizing the following using a least squares optimization method.

  E(x, y, σ) = (DoG(x, y, σ) - D(x, y, σ) * (x, y, σ))^2

- **Orientation assignment:** The orientation of each key point is determined based on the dominant gradient directions in the image patch around the key point. The gradient of the image at each point (x, y) is defined as:

  ∇I(x, y) = [Ix(x, y), Iy(x, y)]

  where Ix(x, y) and Iy(x, y) are the partial derivatives of the image with respect to x and y, respectively. The orientation of the key point is determined by computing the histogram of gradient orientations in the image patch around the key point and selecting the orientation with the highest peak as the dominant orientation.

- **Keypoint descriptor:** A descriptor is then created for each keypoint, which is a vector of values that describes the appearance of the image patch around the keypoint. . The keypoint descriptor is used to represent the distinctive features of the keypoint, and it can be used to identify and match the keypoint with other key points in different images.

  To create a keypoint descriptor, the SIFT algorithm divides the image patch around the keypoint into smaller cells and computes histograms of gradient orientations in each cell. The gradient orientations are computed using the following equation:

  orientation = atan2(Iy, Ix)

  where Ix and Iy are the partial derivatives of the image with respect to x and y, respectively. The histograms are weighted by the magnitude of the gradient and the Gaussian weighted distance from the center of the keypoint. The resulting histograms are concatenated to form the keypoint descriptor, which is a vector of values that describes the appearance of the image patch around the keypoint. The keypoint descriptor is used to identify and match key points between different images. Remove low contrast key points (key point selection)

d. Feature matching

Feature matching is the process of identifying corresponding points in two or more images that belong to the same object or scene. It is an important step in many computer vision tasks, such as object recognition, image registration, and structure from motion. There are various ways to perform feature matching, but one common approach is to extract distinctive features from each image, and then use these features to establish correspondences between the images.

G2NN(Geometric Two Nearest Neighbor): Geometric Two Nearest Neighbor (g2nn) is a feature matching algorithm used in computer vision and image processing. It is used to identify and match similar feature

Keypoint description extraction: The first step in using the G2NN algorithm is to extract descriptions of the key points in the image. In the provided code, this is done using the SIFT algorithm, which extracts numerical descriptions of the key points based on the gradient orientation of pixels in a small patch around the keypoint.

Let the set of keypoints in the image be denoted as

K = {k1, k2, ..., kn},

the set of descriptions for these key points be denoted as

D = {d1, d2, ..., dn}.

Euclidean distance calculation: Once descriptions have been extracted from the keypoints, the next step is to calculate the Euclidean distance between each pair of keypoints

d(k1,k2) is the distance them keypoint k1 and k2 d(k1,k3)) is the distance them keypoint k1 and k3

the Euclidean distance formula is given by:

$$d = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}$$

Where,

"d" is the Euclidean distance

(x_1, y_1) is the coordinate of the first point

(x_2, y_2) is the coordinate of the second point

After calculating the Euclidean distance between all pairs of keypoints, the next step in the g2nn1 function is to sort the distances in ascending order.

Once the keypoint pairs have been sorted by distance, the G2NN algorithm compares the distances between each pair of keypoints to a predetermined threshold value, which is specified as the global variable G2NN_THRESHOLD. If the ratio of the distance between a pair of keypoints to the distance between the next closest pair of keypoints is greater than the threshold value, the G2NN algorithm considers the keypoints to be a match and adds them to a list of matches.

For example, if the keypoint pairs were sorted by distance as follows:

(k1, k2), d(k1,k2) = 0.5

(k1, k3), d(k1,k3) = 0.7

(k1, k4), d(k1,k4) = 5

And if the threshold value was set to 0.5, the G2NN algorithm would consider the keypoints k1 and k2 to be a match, because the ratio of d(k1,k2)/d(k1,k3) is greater than the threshold value. However, the keypoints k1 and k3 would not be considered a match, because the ratio of d(k1,k3)/d(k1,k4) is less than the threshold value.

Image after feature extraction and matching

e. Clustering

Clustering is a technique that groups together similar data points into clusters. The goal of clustering is to partition the data points into groups such that the points within each group are more similar to each other than to points in other groups. In the project , the Hierarchical Clustering

algorithm is used to group together similar points into clusters.

Hierarchical Clustering: Initialize the clusters: Each data point is initially its own cluster. Calculate the distance between all pairs of clusters: The distance between two clusters is typically calculated as the Euclidean distance between their centroids, which are the average positions of the points in each cluster. Merge the two closest clusters: Create a new cluster that includes all the patches in the two closest clusters. The centroid of the new cluster can be calculated as the average position of the patches in the cluster**.** Update the clusters: Remove the two merged clusters and add the new merged cluster to the set of clusters.

Repeat steps until there are only two clusters left or until the distance between the two closest clusters is greater than the threshold: The algorithm enters a loop that repeats the above steps until the condition is met. When the loop is finished, the function returns the final set of clusters. Identify forgeries: Regions with a large number of clusters or clusters that are spatially close to each other are more likely to contain copies and can be flagged as potential forgeries.

After clustering

Final output copy and move forgery is detected

CONCLUSION

The outcomes of our experiments affirm the remarkable efficacy of our algorithm in detecting copy-move forgery, even in the presence of various forms of noise and interference. In the ever-evolving landscape of digital media, the challenges posed by image manipulation and forgery persist and evolve as well. As digital technology advances, so do the methods and tools available to those who engage in forgery. These individuals constantly seek innovative ways to alter or manipulate images, leaving behind subtler traces and making their actions harder to detect.

In light of this ongoing challenge, it becomes imperative to implement enhanced security measures and continually advance our detection techniques. The integrity of digital content is of paramount importance, especially in applications such as journalism, where trustworthy visual evidence is crucial. Additionally, businesses and individuals rely on accurate and unaltered images for marketing, personal communication, and a range of other purposes. Thus, the demand for effective security and detection

mechanisms is on the rise, driven by the need to ensure the authenticity and reliability of digital media in an environment characterized by constant change and sophistication.

As the digital landscape continues to evolve, it is crucial to stay at the forefront of research and innovation in the field of digital image forensics. This allows us to develop and adapt detection methods that can effectively counter the ever-advancing techniques employed by those who engage in image forgery. By doing so, we can better protect the integrity of digital content and maintain trust in the images that play a vital role in our modern society.

REFERENCES

A. J. FRIDRICH, B. D. SOUKAL, AND A. J. LUKÁŠ, "DETECTION OF COPY-MOVE FORGERY IN DIGITAL IMAGES," IN PROCEEDINGS OF DIGITAL

FORENSIC RESEARCH WORKSHOP, 2003.[2] A. C. POPESCU AND H. FARID, "EXPOSING DIGITAL FORGERIES BY DETECTING

DUPLICATED IMAGE REGIONS," DEPT. COMPUTER. SCI., DARTMOUTH COLLEGE,TECH. REP. TR2004-515, 2004.

LOWE, DAVID G. (1999). "OBJECT RECOGNITION FROM LOCAL SCALE-INVARIANT FEATURES" (PDF). PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON COMPUTER VISION. VOL. 2. PP. 1150–1157. DOI:10.1109/ICCV.1999.790410

POPESCU A. C, FARID. H, "EXPOSING DIGITAL FORGERIES BY DETECTING TRACES OF RE-SAMPLING," IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 53, NO. 2,2005, PP. 758-767.

QIONG WU, GUO-HUI LI, SHAO-JIE SUN, "DETECTION OF COPY FORGERY REGIONS IN THE IMAGE BASED ON WAVELET AND SINGULAR VALUED COMPOSITION," JOURNAL OF CHINESE COMPUTER SYSTEMS, NO. 4, 2008.

POOJA BHOLE, DIPAK WAJGI, 2020, AN IMAGE FORGERY DETECTION USING SIFT-PCA, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY