



**End term T12 - Project Phase II  
on**

SPYHUB- Multiple techniques to detect Copy move Image Forgery

**Submitted by**

**Project Members**

Aditya Dhenge 1032180067  
Palak Mallwat 1032181321  
Rashi Madnani 1032181247  
Dhanashree Lodhe 103218251

**Under the Internal Guidance of**

Prof. Sajeeda Shikalgar

**Under the External Guidance of (if applicable)**

**School of Computer Engineering and Technology  
MIT World Peace University, Kothrud,  
Pune 411 038, Maharashtra - India  
2020-2021**



Dr. Vishwanath Karad

**MIT WORLD PEACE  
UNIVERSITY** | PUNE

TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

**SCHOOL OF COMPUTER ENGINEERING AND TECHNOLOGY**

**C E R T I F I C A T E**

This is to certify that, Aditya Dhenge 1032180067, Palak Mallwat 1032181321, Rashi Madnani 1032181247, Dhanashree Lodhe 103218251 of BTech.( Computer Science & Engineering) have completed their project titled “Mention Project title here” and have submitted this Capstone Project Report towards fulfillment of the requirement for the Degree-Bachelor of Computer Science & Engineering (BTech-CSE) for the academic year 2020-2021.

[Dr/ Prof.]  
Project Guide  
School of CET  
MIT World Peace University, Pune.  
Pune

[Dr. Vrushali Kulkarni]  
Program Head  
School of CET  
MIT World Peace University,

Internal Examiner:

External Examiner:

Date:

# Acknowledgement

It gives us great pleasure in presenting the capstone project report on SPYHUB- Multiple techniques to detect Copy-Move Image Forgery.

We would like to take this opportunity to thank our internal guide Prof. Sajeeda Shikalgar for giving us all the help and guidance we needed. We are really grateful to her for her kind support. Her valuable suggestions were very helpful.

We are also grateful to Prof. Jayshree Aher, External guide for her indispensable support and suggestions.

Aditya Dhenge  
Palak Mallwat  
Rashi Madnani  
Dhanashree Lodhe

## **Abstract**

With the rapid development of the Internet, it becomes easy to obtain abundant multimedia information. It is convenient for people to get high-resolution pictures and videos with their cameras or mobile phones, enriching their lives. However, people can alter the content of images as their wishes using various image editing software arbitrarily, such as Adobe PhotoShop and ACDSee Photo Editor. The authenticity and integrity of images have been threatened in many critical fields. Therefore, image forensics technique as a significant part of information security, which aims at identifying the forgery, is urgent to be developed.

## List of Figures

<b>fig.No</b>	<b>Figure Name</b>	<b>Page No.</b>
Fig.1.	Methods to detect copy- move forgery	9
Fig.2.	DFD level 0	17
Fig.3.	DFD level 1	17
Fig.4.	DFD level 2	18
Fig.5.	Block diagram	18
Fig.6.	Use case diagram	19
Fig.7.	Activity Diagram	19
Fig.8.	Sequence Diagram	19
Fig.9.	State Transition Diagram	20
Fig.10.	Difference of Gaussian Kernel	23
Fig.11.	Pixels Representation	23
Fig.12.	16x16 window & 128-dimensional vector	25
Fig.13.	Gaussian partial derivative in xy	27
Fig.14.	Gaussian partial derivative in y	27
Fig.15.	Orientation Assignment	28
Fig.16.	descriptor components	28
Fig.17.	test case 1	30
Fig.18.	test case 2	30
Fig.19.	test case 3	31
Fig.20.	test case 4	31
Fig.21.	Result of test case 1	32
Fig.22.	Result of test case 2	32
Fig.23.	Result of test case 3	33
Fig.24.	Result of test case 4	33

## List of Tables

Table.No	Table Name	Page No.
Table.1.	Literature Survey	10
Table.2.	Requirements Rationale	15
Table.3.	Risk Management	15
Table.4.	RRM Table	16
Table.5.	Project Plan	20

# Index

Sr.No	Content	Page.No
1.	Introduction	8
	1.1 Project Statement	8
	Area	8
	Project Introduction and Aim	8
2.	Literature Survey	10
3.	Problem Statement	13
	3.1 Project Scope	13
	3.2 Project Assumptions	13
	3.3 Project Limitations	13
	3.4 Project Objectives	14
4.	Project Requirements	14
	4.1 Resources	14
	4.2 Software & Hardware requirements	14
	4.3 Requirements Rationale	15
	4.4 Risk Management	15
	4.5 Functional Specifications	16
	4.5.1 Interfaces	16
5.	System Analysis Proposed Architecture/high-level design of the project	17
	5.1 Design Consideration	17
	5.2 Block Diagram	18
	5.3 UML Diagram	19
	5.3.1 Use Case Diagram	19
	5.3.2 Activity Diagram	19
	5.3.3 Sequence Diagram	19
	5.3.4 State Transition Diagram	20
6.	Project Plan	20
7.	Implementation	21
	7.1 Methodology	21
	7.2 Algorithm	21
	7.3 Other Implementation details	29
8.	Performance Evaluation and Testing	29
	8.1 Discuss various test plans	29
9.	Result and Analysis	31
	9.1 Explanation: how experiment has been performed	31
	9.2 Discuss Results	32
10.	Applications	33
11.	Conclusion	33
12.	Future prospects of the project	34
13.	References	35
14.	Appendices	36

# 1. Introduction

## 1.1 Project Statement

SPYHUB- Multiple techniques to detect Copy move Image Forgery

## 1.2 Area:

Machine Learning

## 1.3 Project Introduction and Aim :

With the rapid development of the Internet, it becomes easy to obtain abundant multimedia information. It is convenient for people to get high-resolution pictures and videos with their cameras or mobile phones, enriching their lives. However, people can alter the content of images as their wishes using various image editing software arbitrarily, such as Adobe PhotoShop and ACDSee Photo Editor. The authenticity and integrity of images have been threatened in many critical fields. For example, forged medical films may cause misdiagnosis and affect the state of illness, and forged newspaper photographs may mislead people and cause social turbulence. Therefore, image forensics technique as a significant part of information security, which aims at identifying the forgery, is urgent to be developed.

In recent decades, scholars have proposed different methods to distinguish between original images and forgery images, which are divided into active forensics and passive forensics. Active forensics techniques are used to verify the integrity of the verification information such as digital watermark and digital signature. Active forensics techniques have the advantages of strong detection ability and are not easy to be avoided. However, in active forensics techniques, the verification information needs to be inserted into carrier images before distribution, which decreases the quality of images. Passive forensics techniques are used to verify the authenticity by analyzing the information and structure of images, which overcome the defects of active forensics techniques. There are mainly two forgeries to alter the content of images: splicing and copy-move. Splicing forgery is a way to copy and paste a part of an image into another image. Copy-move forgery is a way to copy and paste a part of an image into the same image.

Copy-move forgery detection techniques are of the following three types:-

- Brute Force
- Block-Based Techniques
- Keypoint Based Techniques



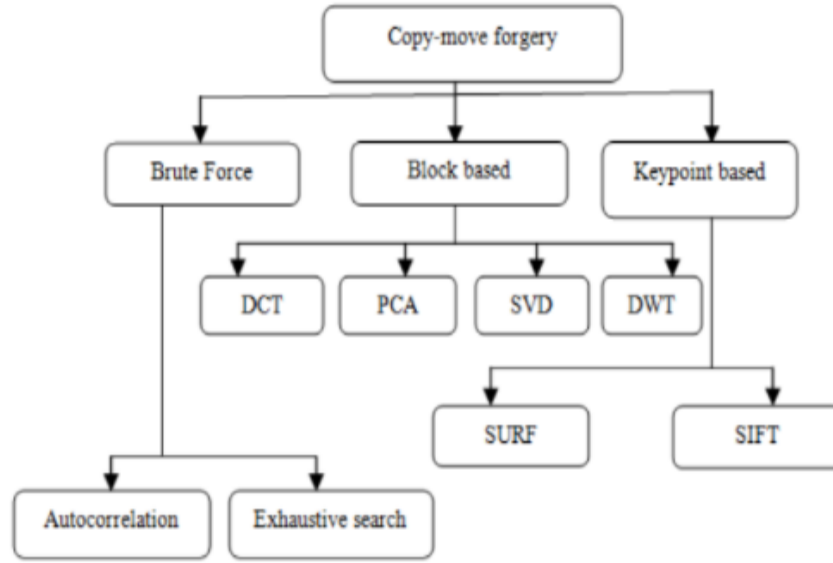


Fig 1: Methods to detect copy- move forgery

All the existing systems involving brute force methods are based on exhaustive search and autocorrelation techniques. In an exhaustive search, the image is used to examine matching segments with circularly shifted versions. As it makes such a large number of comparisons, its computational unpredictability is high.

In the block-based techniques the whole image is broken up into small blocks and suitable features are extracted from each block, then finally these features are matched to predict the forgery. This raises two problems, it is difficult to determine the block size suitable for the given image. If the block size is too big, it would fail to detect small copy and move forgeries and if the block size is small, the model will become too complex.

The keypoint-based forgery detection techniques have shown good performance specially when geometric transformations (e.g. rotation and scaling) are applied, because of rotation and scaling invariant properties of keypoint extractor. Also, the keypoint-based techniques are found to be very robust in detecting forgery when an image is post-processed using JPEG compression, noise addition, brightness change etc. The keypoint-based techniques are comparatively less time demanding.

Thus, in our project we plan to implement keypoint based methods such as SIFT and SURF for feature extraction. Following that, feature matching will be used to find similar parts in a picture. We utilize brute force match followed by knn match to get a list of similar areas in a picture. Outliers must be filtered since feature matching returns a large number of matches. This is accomplished using the RANSAC algorithm, which eliminates the drawbacks of key point-based techniques by distinguishing the genuine forged area and so removing the false matched regions, thereby enhancing the model's accuracy. The system also highlights the places that have been spotted.

## 2. Literature Survey

Table 1: Literature Survey

Sr No	Paper Name	Publication	Characteristics	Research Gap
1.	BRISK and SIFT-based Copy-Move Forgery Detection of Digital Images	2021, International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies	A new methodology for performing CMFD of digital images was developed to improve the computational speed. It employed the BRISK algorithm for identifying the KPs and used SIFT for computing the descriptors at identified KPs. The method used K-means clustering and performed Euclidean distance-based KP matching. It then removed the false KP pairs by employing RANSAC.	Doesn't perform well on pictures with high computational burden.
2.	Copy Move Forgery Detection Using Forensic Images	2021, Iraqi Journal of Science, Vol. 62, No. 9, pp: 3167-3181 DOI: 10.24996/ijs.2021.62.9.31	Scale invariant feature transform (SIFT) descriptor is applied The forgery detection results gave a performance percent of about 98% The accuracy of detection is improved by raising the value of the radius until the optimum one is reached when the radius is Three pixels, giving a mean detection value of approximation 98 %.	The change in image resolution leads to decrease the accuracy of the segmentation results, which affects the features extraction by SIFT descriptor. The measured average processing time for implementation was about (3-4) hours.
3.	Copy-Move Forgery Detection Based on Keypoint	2020, IEEE Access Special Section On Innovation And Application Of Internet Of Things And	CMFD method via clustering SIFT keypoints The proposed method is superior to existing state-of-art methods in terms of	In large-scale forgery, this method is not stable and has poor effects, which is also

	Clustering and Similar Neighborhood Search Algorithm	Emerging Technologies In Smart Sensing	matching time complexity, detection reliability, and forgery location accuracy. The method has better robustness and could accurately locate tampered regions, especially on simple forgery, geometric transformation forgery, and small-scale post-processing forgery.	the problem we need to solve in the future.
4.	A hybrid copy-move image forgery detection technique based on Fourier-Mellin and scale invariant feature transforms	2020, Springer Science+Business Media, LLC, part of Springer Nature	This paper proposes a robust hybrid copy-move forgery detection technique using the Fourier-Mellin transform (FMT) and SIFT algorithm. The results also show that the proposed technique is comparatively less time demanding. Furthermore, the proposed technique works very well even in extreme conditions like scaling with factor 50%–200%, and JPEG compression with quality factor up to 20.	Doesn't perform well under scaling factor in the range [0.8–1.2] and rotation compared to the existing CMFD techniques.
5.	Image matching based on the adaptive redundant keypoint elimination method in the SIFT algorithm	2020, Springer-Verlag London Ltd., part of Springer Nature 2020	An adaptive algorithm for eliminating redundant keypoints in the SIFT method was presented. The proposed algorithm, generated based on the RKEM–SIFT method, enhanced the matching performance with respect to its origin in terms of different indices. One of the main characteristics of the	Existing numerous redundant key points located very close to each other in the image. These redundant key points increase the computational complexity while they decrease the image matching performance.

			proposed adaptive RKEM method was that the threshold value could be chosen differently in the sensed and reference images unlike the RKEM.	
6.	An Image Copy-Move Forgery Detection Method Based on SURF and PCET	2019, IEEE Access	Aiming at the difficulty of detecting forgery which occurs in high-brightness smooth regions or forgery images A method based on SURF and PCET is proposed. The proposed method combines the advantages of block-based and keypoint-based image CMFD methods.	The parameters are difficult to generalize in various conditions. In feature matching section,a time-consuming method is used In addition, due to the sampling or interpolation of large-scale reduction or enlargement in the image regions, it is difficult to detect whether the images are tampered
7.	A hybrid copy-move image forgery detection technique based on Fourier-Mellin and scale invariant feature transforms	Symmetry 2018	In this paper, a CMFD scheme based on A-KAZE and SURF was proposed. To obtain sufficient points in the smooth regions, the response thresholds for A-KAZE and SURF were set to small values instead of their default parameters. In particular, a new correlation map was presented in this paper that can demarcate the duplicated regions with closed regions in tampered images	When the tampered region is distorted by image manipulation, this may not be as effective as that in plain image copy-move forgery detection. Compared to other tested CMFD methods, the proposed method exhibits both advantages and disadvantages.
8.	Image forgery detection for high resolution images	2017, Proceedings of the 2nd International Conference on Communication and	The proposed algorithm mainly involves matching the tentacles of the same features extracted from	Dealt only with high resolution images and had the same recall

	using SIFT and RANSAC algorithm	Electronics Systems (ICCES 2017) IEEE Xplore Compliant - Part Number:CFP17AWO-ART, ISBN:978-1-5090-5013-0	each block by computing the dot product between the unit vectors. The RANSAC algorithm is applied to detect the forged regions. The proposed method has 97% precision This scheme giving more accurate results and performances proves to be a novel technique to be implemented	rates as compared to prev proposed methods
--	---------------------------------	---	---	--

### 3. Problem Statement

#### 3.1 Project Scope

Following are the functionality that are implemented in the proposed model:

1. Image resizing and preprocessing using python libraries
2. Feature Extraction using either SIFT or SURF
3. Feature Matching using Brute Force Matcher
4. Filtering Outliers using RANSAC
5. Forgery Detection and Highlighting the manipulated areas
6. Saving the results.

#### 3.2 Project Assumptions

1. The user will upload an image to test.
2. The project will be focussing on detection of Copy-Move Forgery.
3. The image size should be small (<1 Mb).

#### 3.3 Project Limitations

The prime drawback of the existing methods is Automation. This means that the answers can be interpreted with the intervention of humans only and does not really need automation.

Secondly drawback is that if we talk about copy-move forgery, then the use of these methods is computationally expensive. Thirdly, at present there is no technique which can identify between the malicious forgery and just the retouching like artistic manipulation.

The most challenging task is to develop a unified algorithm having capacity to detect any type of forgery.

### 3.4 Project Objectives

Our project will begin by allowing the user to input an image in which they want to detect the copy-move forgery (CMF). They will further be given an option of selecting which method they want to employ for Copy and move forgery detection, namely (i) Scale invariant Feature Transform (SIFT) and (ii) Speeded Up Robust Features (SURF) .

The selected model will then detect the forged areas in the image given by the user to test. Our system will also highlight the forged areas in the given image. The user will then have an option to save the image with highlighted forgery detected areas. The user can then upload another image or close the program.

## 4. Project Requirements

### 4.1 Resources

- Human Resources
  - Team of 4 capstone team members, Capstone guide by college, External Evaluators.
- Reusable Software Components
  - Feature matching: A brute-force matcher is a descriptor matcher that compares two sets of keypoint descriptors and generates a result that is a list of matches. It is called brute-force because little optimization is involved in the algorithm.
  - Filtering outliers: Random sample consensus, or RANSAC, is an iterative method for estimating a mathematical model from a data set that contains outliers. The RANSAC algorithm works by identifying the outliers in a data set and estimating the desired model using data that does not contain outliers.

### 4.2 Software & Hardware requirements

- Hardware Requirements:
  - Intel i5/i7 processor
  - RAM 8GB
  - GPU 8GB RAM
  - PC Or server with x86-64(64 bit)
- Software Requirements:
  - Spyder IDE
  - Operating system-Windows 10/11

#### 4.3 Requirements Rationale

Table 2: Requirements Rationale

Requirements	Rationale
Processor	Used for the processing requirements
Hard Disk	Used to store and interface the relevant data
RAM	For the purpose of fulfilling the volatile memory requirements
Operating System	For enabling the operation of the methodology
Development Kit	For the effective libraries and functionality of the programming language
IDE	For coding purposes in the python language
GUI	For interfacing with the user

#### 4.4 Risk Management

Following are the details for each risk:

Table 3: Risk Management

Probability	Value	Description
High	Probability of occurrence is	>75%
Medium	Probability of occurrence is	26-75%
Low	Probability of occurrence is	<25%

Table 4 : RRM Table

Risk	Risk type	Probability Of occurrence	impact	priority	Rmmm (Risk management)	plan
SYSTEM FAILURE	Technical	30%	3-critical	high	Daily based monitoring:-By serviced employees	Modifying using software developers on serious issues
CORRUPT ED IMAGE	Module specific	60%	4-marginal	high	By adding layers to distinguish the corrupted files	Reduction or discarding if required
VERSION ISSUE	Technical	10%	3-medium	Medium	By changing the versions and integrating new model versions	Keeping track of the working of the versions and updating it.

#### 4.5 Functional Specifications:

##### 4.5.1 Interfaces

- External Interfaces Required
  - Our system interacts with user on the following occasions:
    1. By the user while inputting the image.
    2. By the user while viewing and saving the results.
- Communication Interfaces
  - Our system's different modules are communicating with one another on the following scenarios:
    1. From reading input image to preprocessing module.
    2. From preprocessing module to selecting the feature extraction module.
    3. From feature extraction module to Brute Force Matching module.
    4. From Brute Force Matching module to Filtering outliers module.
    5. From Filtering outliers module to Display prediction Module.



## 6. From Display prediction module to Plot the forgeries Module.

- Graphical User Interfaces
  - Our system uses Tkinter graphical user interface (GUI) framework of python.

## 5. System Analysis Proposed Architecture/high-level design of the project

### 5.1 Design Consideration

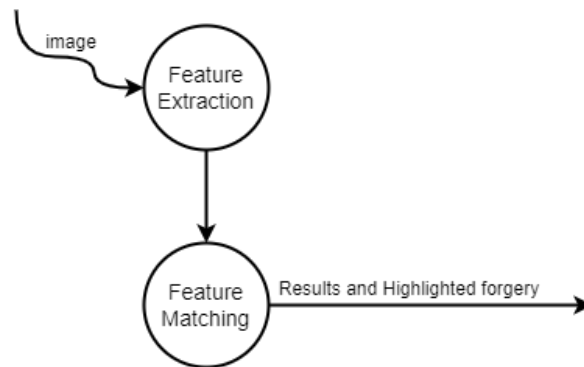


Fig 2: DFD level 0

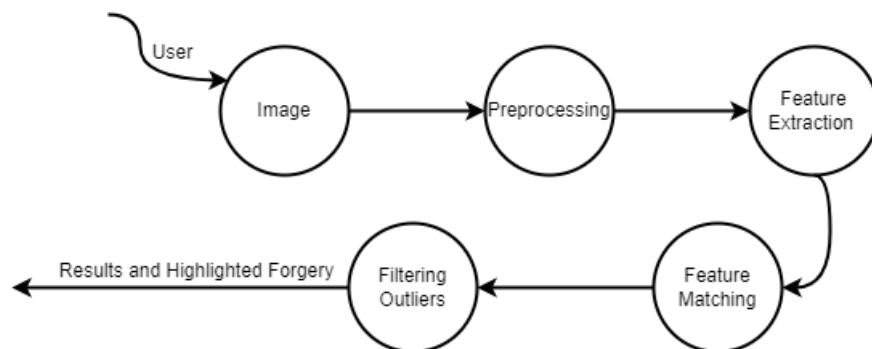


Fig 3: DFD level 1

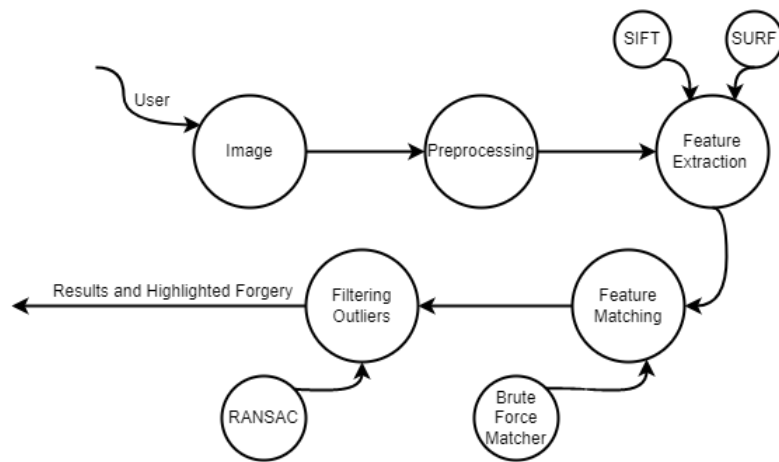


Fig 4: DFD level 2

## 5.2 Block Diagram

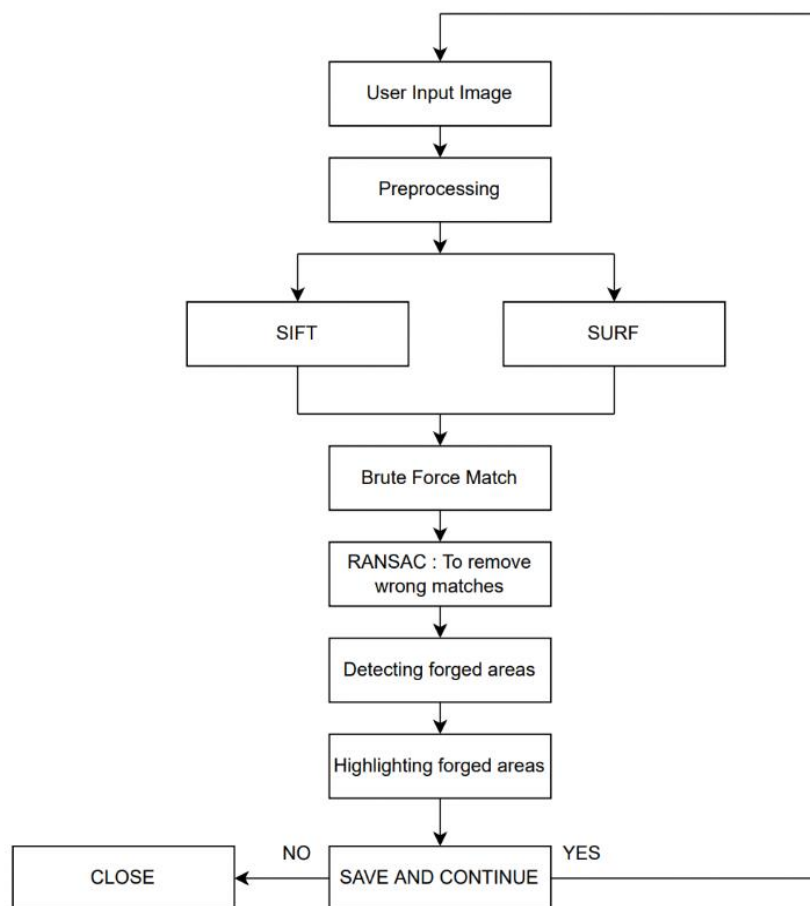


Fig 5: Block diagram

## 5.3 UML Diagram

### 5.3.1 Use Case Diagram

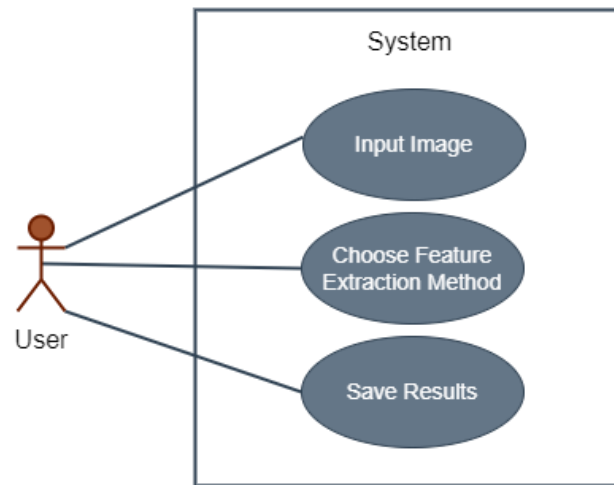


Fig 6: Use case diagram

### 5.3.2 Activity Diagram

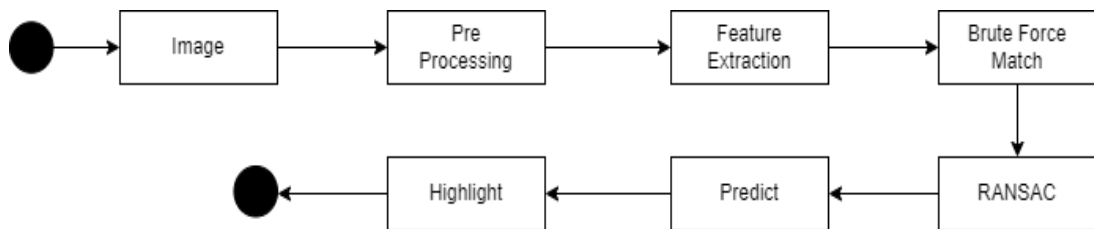


Fig 7: Activity Diagram

### 5.3.3 Sequence Diagram

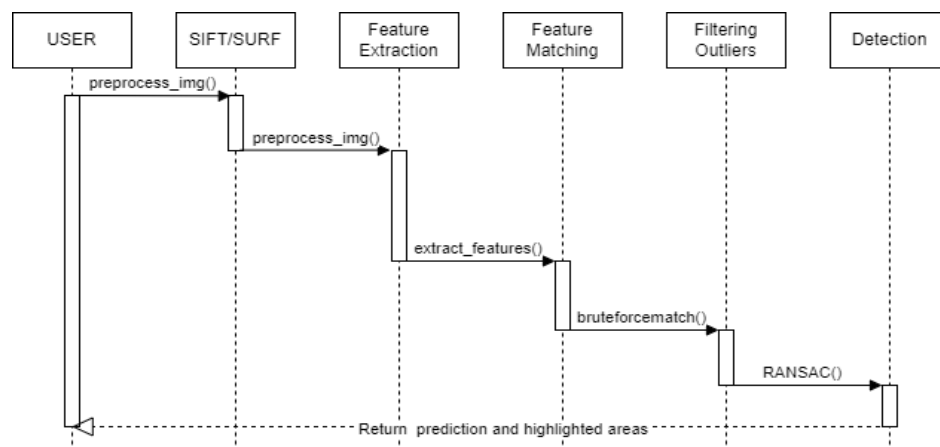


Fig 8: Sequence Diagram

### 5.3.4 State Transition Diagram

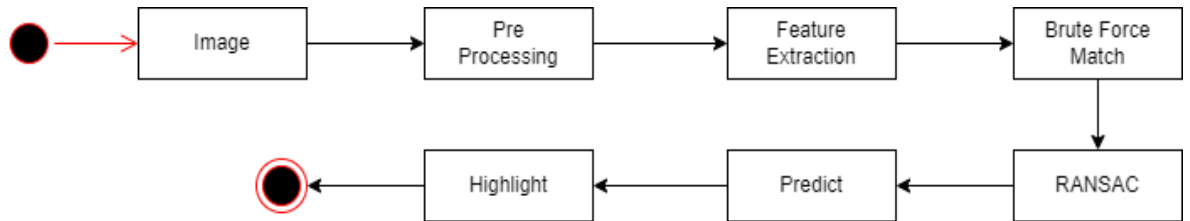


Fig 9: State Transition Diagram

## 6. Project Plan

Table 5: Project Plan

Activity Month	Year 2021		Year 2022				
	Nov	Dec	Jan	Feb	Mar	Apr	May
Literature Survey and Review, Synopsis submission	✓	✓					
Literature Survey Finalization		✓					
Proposed Model Analysis		✓	✓				
Report Writing and Paper Writing/Presentation/Publication			✓	✓			
SIFT coding			✓	✓			
SURF and RANSAC					✓	✓	
User Interface						✓	✓

## 7. Implementation

### 7.1 Methodology :

1] Brute Force : A brute force approach is an approach that finds all the possible solutions to find a satisfactory solution to a given problem. The brute force algorithm tries out all the possibilities till a satisfactory solution is not found. Brute Force Algorithms are exactly what they sound like – straightforward methods of solving a problem that rely on sheer computing power and trying every possibility rather than advanced techniques to improve efficiency.

2] Ransac : Random sample consensus (RANSAC) is an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers, when outliers are to be accorded no influence on the values of the estimates. Therefore, it also can be interpreted as an outlier detection method. It is a non-deterministic algorithm in the sense that it produces a reasonable result only with a certain probability, with this probability increasing as more iterations are allowed. The algorithm was first published by Fischler and Bolles at SRI International in 1981. They used RANSAC to solve the Location Determination Problem (LDP), where the goal is to determine the points in the space that project onto an image into a set of landmarks with known locations.

3] SIFT : Scale-invariant feature transform (SIFT) is a broadly adopted feature extraction method in image classification tasks. The feature is invariant to scale and orientation of images and robust to illumination fluctuations, noise, partial occlusion, and minor viewpoint changes in the images. These characteristics are important for mitosis detection when cells in the image have different sizes and orientations. The SIFT feature is composed of several key points in the image with an orientation and the corresponding descriptor of the area around the selected key points. SIFT key points are searched through different image scales, known as the Difference of Gaussian (DoG) pyramid.

4] SURF : SURF is the speed up version of SIFT. In SIFT, Lowe approximated Laplacian of Gaussian with Difference of Gaussian for finding scale-space. SURF goes a little further and approximates LoG with Box Filter. One big advantage of this approximation is that, convolution with box filter can be easily calculated with the help of integral images. And it can be done in parallel for different scales. Also, the SURF rely on determinant of Hessian matrix for both scale and location. For orientation assignment, SURF uses wavelet responses in horizontal and vertical direction for a neighborhood of size 6s. Adequate gaussian weights are also applied to it. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window of angle 60 degrees. wavelet response can be found out using integral images very easily at any scale. SURF provides such a functionality called Upright-SURF or U-SURF. It improves speed and is robust upto . OpenCV supports both, depending upon the flag, upright. If it is 0, orientation is calculated. If it is 1, orientation is not calculated and it is faster.

### 7.2 Algorithm :

1] There are mainly four steps involved in the SIFT algorithm. We will see them one-by-one.

- **Scale-space peak selection:** Potential location for finding features.

- **Keypoint Localization:** Accurately locating the feature keypoints.
- **Orientation Assignment:** Assigning orientation to keypoints.
- **Keypoint descriptor:** Describing the key-points as a high dimensional vector.

### Phase 1: Scale-space (Potential location for finding features)

Only at a certain scale do real-world objects have any relevance. A sugar cube may appear to be properly placed on a table. When gazing at the Milky Way as a whole, though, it simply does not exist. Objects with many scales are fairly frequent in nature. A scale space is a computer graphic that attempts to emulate this concept. The scale space of an image is a function  $L(x,y,\sigma)$  obtained by convolutioning a Gaussian kernel (Blurring) with the input image at various scales. The number of octaves and scale used in scale-space is determined by the size of the original image. As a result, we create many octaves of the original image. The image size of each octave is half that of the previous one. Using the Gaussian Blur operator, images are gradually blurred across an octave.

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$$

The convolution of the Gaussian operator with the image is referred to as "blurring" in mathematics. Each pixel in a Gaussian blur has a unique expression or "operator" applied to it. The outcome is a smudged image.

The Gaussian Blur operator is  $G$ , and the image is  $I$ . The location coordinates are  $x,y$ , and the "scale" parameter is  $\sigma$ . Consider it the amount of blur. The blur is increased as the value increases.

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2 + y^2)/2\sigma^2}$$

Now we'll utilize those blurred images to create a new collection of photographs called the Gaussian Difference. The scale space of an image is a function  $L(x,y,\sigma)$  obtained by convolutioning a Gaussian kernel (Blurring) with the input image at various scales. The number of octaves and scale used in scale-space is determined by the size of the original image. As a result, we create many octaves of the original image. The image size of each octave is half that of the previous one. Using the Gaussian Blur operator, images are gradually blurred across an octave.

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$$

The convolution of the Gaussian operator with the image is referred to as "blurring" in mathematics. Each pixel in a Gaussian blur has a unique expression or "operator" applied to it. The outcome is a smudged image.

The Gaussian Blur operator is  $G$ , and the image is  $I$ . The location coordinates are  $x, y$ , and the "scale" parameter is  $\sigma$ . Consider it the amount of blur. The blur is increased as the value increases.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2 + y^2)/2\sigma^2}$$

Now we'll utilize those blurred images to create a new collection of photographs called the Gaussian Difference (DoG). These DoG (Fig 2) pictures are excellent for identifying noteworthy keypoints in a photograph. The difference of Gaussian is obtained as the difference of Gaussian blurring of an image with two different  $\sigma$ , let it be  $\sigma$  and  $k\sigma$ . In the Gaussian Pyramid, this process is repeated for different octaves of the image.

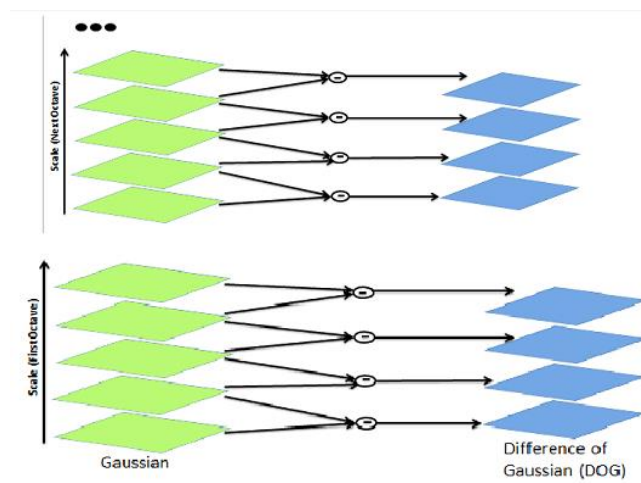


Fig 10: Difference of Gaussian Kernel

We've created a scale space and utilized it to calculate the Difference of Gaussians up to this point. These are then utilized to calculate scale-invariant Laplacian of Gaussian approximations.

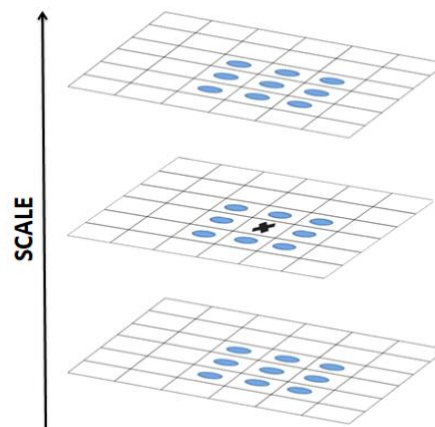


Fig 11: Pixels Representation

In an image, one pixel (Fig 3) is compared to its 8 neighbors, as well as 9 pixels in the next scale and 9 pixels in the following scale. A total of 26 checks are performed in this manner. It's a potential keypoint if it's a local extrema. It basically says that that scale best represents the key-point.

## B. Phase 2: Keypoint Localization

The keypoints created in the preceding phase result in a large number of keypoints. Some of them are too close to the edge, or there isn't enough contrast. They aren't as valuable as features in both circumstances. As a result, we get rid of them.

The method is similar to that used to remove edge features in the Harris Corner Detector. We just assess the intensities of low contrast features. They employed a Taylor series expansion of scale space to establish a more precise location of extrema, and if the intensity at these extrema is less than a threshold value (0.03, according to the research), it is rejected.

Edges have a stronger response in DoG, so they must be removed as well. The primary curvature was calculated using a 2x2 Hessian matrix (H).

Reject Flags:

$$|D(\hat{x})| < 0.03$$

Reject

Edges:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

Let  $\alpha$  be the eigenvalue with larger magnitude and  $\beta$  the smaller.

$$\text{Tr}(H) = D_{xx} + D_{yy} = \alpha + \beta,$$

$$\text{Det}(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

Let  $r = \alpha/\beta$

So,

$$\alpha = r\beta$$

$$\frac{\text{Tr}(H)^2}{\text{Det}(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r},$$

$(r+1)^2/r$  is at a min when the 2 eigenvalues are equal.

## C. Phase 3: Orientation Assignment

We now have verifiable keypoints. They've been thoroughly tested for stability. The scale at which the keypoint was detected is already known (it's the same as the blurred image's scale). As a result, scale invariance exists.

The next step is to give each keypoint an orientation to ensure rotation invariance. Depending on the scale, a neighborhood is drawn around the keypoint location, and the gradient magnitude and direction are determined in that area.

The result is a 360-degree orientation histogram with 36 bins. If the gradient direction at a given position (in the "orientation collecting zone") is 18.759 degrees, it will be classified as 10–19 degrees. And the



"quantity" that goes into the bin is proportional to the size of the bin. the gradient's magnitude at that point The histogram will have a peak at some point after you've done this for all pixels surrounding the keypoint.

To compute the orientation, the highest peak in the histogram is used, as well as any peak above 80% of it. It creates keypoints that are the same size and location, but face in different directions. It contributes to the matching's stability.

#### D. Phase 4: Keypoint Descriptor

Each keypoint now has a location, scale, and orientation. The next step is to create a descriptor for each keypoint's local picture region that is extremely unique and as invariant as feasible to changes in viewpoint and illumination. To accomplish so, a 16x16 window (Fig 4) is created around the keypoint. It's broken down into 16 4x4 sub-blocks.

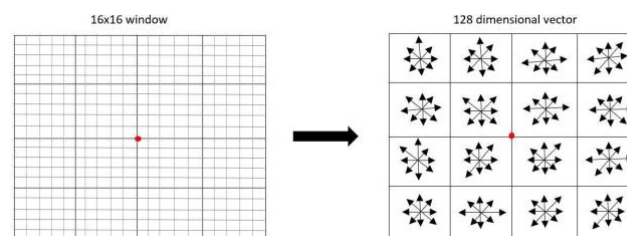


Fig 12: 16x16 window & 128-dimensional vector

In practice, 4x4 descriptors were employed over a 16x16 sample array. The directions 4x4x8 yield 128 bin values. To form a keypoint descriptor, it is expressed as a feature vector. There are a few issues with this feature vector. Before we can finish the fingerprint, we need to get rid of them.

Rotational sensitivity: Gradient orientations are used in the feature vector. Everything changes when you rotate the image, as you can see. The orientations of all gradients shift as well. The rotation of the keypoint is removed from each orientation to achieve rotation independence. As a result, each gradient orientation is relative to the orientation of the keypoint.

Lighting is a factor: We can attain illumination independence by using large threshold numbers. As a result, any integer bigger than 0.2 (out of 128) gets transformed to 0.2.

This feature vector is then normalized one more time. You now have a feature vector that is independent of illumination!

2] SURF Algorithm is composed of two steps :

- Feature Extraction
- Feature Description

#### 1] Feature Extraction

The approach for interest point detection uses a very basic Hessian matrix approximation. The Integral Image is used as a quick and effective way of calculating the sum of values (pixel values) in a given

image — or a rectangular subset of a grid (the given image). It can also, or is mainly, used for calculating the average intensity within a given image. They allow for fast computation of box type convolution filters. The entry of an integral image  $I_\Sigma(\mathbf{x})$  at a location  $\mathbf{x} = (x,y)^T$  represents the sum of all pixels in the input image  $I$  within a rectangular region formed by the origin and  $\mathbf{x}$

$$I_\Sigma(\mathbf{x}) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i,j)$$

With  $I_\Sigma$  calculated, it only takes four additions to calculate the sum of the intensities over any upright, rectangular area, independent of its size.

### Hessian matrix-based interest points

Surf uses the Hessian matrix because of its good performance in computation time and accuracy. Rather than using a different measure for selecting the location and the scale (Hessian-Laplace detector), surf relies on the determinant of the Hessian matrix for both. Given a pixel, the Hessian of this pixel is something like:

$$H(f(x, y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

For adapt to any scale, we filtered the image by a Gaussian kernel, so given a point  $\mathbf{X} = (x, y)$ , the Hessian matrix  $H(\mathbf{x}, \sigma)$  in  $\mathbf{x}$  at scale  $\sigma$  is defined as:

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix}$$

where  $L_{xx}(\mathbf{x}, \sigma)$  is the convolution of the Gaussian second order derivative with the image  $I$  in point  $\mathbf{x}$ , and similarly for  $L_{xy}(\mathbf{x}, \sigma)$  and  $L_{yy}(\mathbf{x}, \sigma)$ . Gaussians are optimal for scale-space analysis but in practice, they have to be discretized and cropped. This leads to a loss in repeatability under image rotations around odd multiples of  $\pi/4$ . This weakness holds for Hessian-based detectors in general. Nevertheless, the detectors still perform well, and the slight decrease in performance does not outweigh the advantage of fast convolutions brought by the discretization and cropping.

In order to calculate the determinant of the Hessian matrix, first we need to apply convolution with Gaussian kernel, then second-order derivative. After Lowe's success with LoG approximations (SIFT), SURF pushes the approximation (both convolution and second-order derivative) even further with box filters. These approximate second-order Gaussian derivatives and can be evaluated at a very low computational cost using integral images and independently of size, and this is part of the reason why SURF is fast.

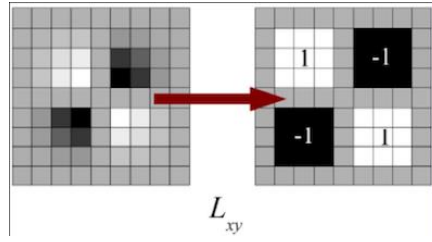


Fig 13: Gaussian partial derivative in xy

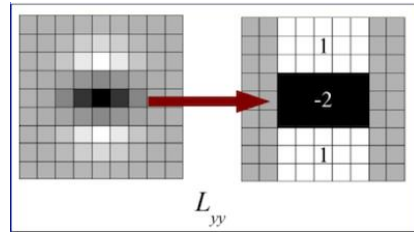


Fig 14: Gaussian partial derivative in y

The  $9 \times 9$  box filters in the above images are approximations for Gaussian second order derivatives with  $\sigma = 1.2$ . We denote these approximations by  $D_{xx}$ ,  $D_{yy}$ , and  $D_{xy}$ . Now we can represent the determinant of the Hessian (approximated) as:

$$\det(\mathcal{H}_{\text{approx}}) = D_{xx}D_{yy} - (wD_{xy})^2, \quad w=0.9 \text{ (Bay's suggestion)}$$

## 2] Feature Description

The creation of SURF descriptor takes place in two steps. The first step consists of fixing a reproducible orientation based on information from a circular region around the keypoint. Then, we construct a square region aligned to the selected orientation and extract the SURF descriptor from it.

### Orientation Assignment

In order to be invariant to rotation, surf tries to identify a reproducible orientation for the interest points. For achieving this:

1. Surf first calculate the Haar-wavelet responses in x and y-direction, and this in a circular neighborhood of radius  $6s$  around the keypoint, with  $s$  the scale at which the keypoint was detected. Also, the sampling step is scale dependent and chosen to be  $s$ , and the wavelet responses are computed at that current scale  $s$ . Accordingly, at high scales the size of the wavelets is big. Therefore integral images are used again for fast filtering.
2. Then we calculate the sum of vertical and horizontal wavelet responses in a scanning area, then change the scanning orientation (add  $\pi/3$ ), and re-calculate, until we find the orientation with largest sum value, this orientation is the main orientation of feature descriptor.

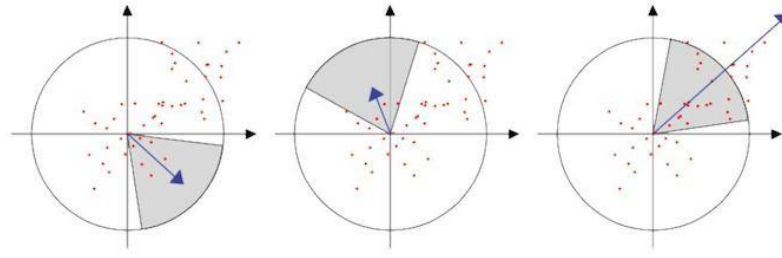


Fig 15: Orientation Assignment

## Descriptor Components

Now it's time to extract the descriptor

1. The first step consists of constructing a square region centered around the keypoint and oriented along the orientation we already got above. The size of this window is  $20s$ .
2. Then the region is split up regularly into smaller  $4 \times 4$  square sub-regions. For each sub-region, we compute a few simple features at  $5 \times 5$  regularly spaced sample points. For reasons of simplicity, we call  $\mathbf{dx}$  the Haar wavelet response in the horizontal direction and  $\mathbf{dy}$  the Haar wavelet response in the vertical direction (filter size  $2s$ ). To increase the robustness towards geometric deformations and localization errors, the responses  $\mathbf{dx}$  and  $\mathbf{dy}$  are first weighted with a Gaussian ( $\sigma = 3.3s$ ) centered at the keypoint.

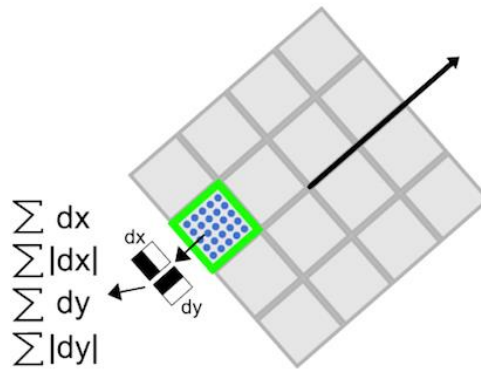


Fig 16: descriptor components

Then, the wavelet responses  $\mathbf{dx}$  and  $\mathbf{dy}$  are summed up over each subregion and form a first set of entries to the feature vector. In order to bring in information about the polarity of the intensity changes, we also extract the sum of the absolute values of the responses,  $|\mathbf{dx}|$  and  $|\mathbf{dy}|$ . Hence, each sub-region has a four-dimensional descriptor vector  $\mathbf{v}$  for its underlying intensity structure  $\mathbf{V} = (\sum \mathbf{dx}, \sum \mathbf{dy}, \sum |\mathbf{dx}|, \sum |\mathbf{dy}|)$ . This results in a descriptor vector for all  $4 \times 4$  sub-regions of **length 64** (In **Sift**, our descriptor is the **128-D vector**, so this is part of the reason that **SURF** is faster than **Sift**).

## 7.3 Other Implementation details

### Feature Matching

Matching detected and reported features entails identifying the correspondence between descriptors in photographs depicting the same item (or scene) from various viewpoints. In this research, a brute-force approach is used to match features between two photographs. On the obtained descriptors, it employs Euclidean distance. Brute Force matching is frequently used in conjunction with the k-Nearest Neighbors method (also known as kNN) and the ratio test to eliminate outliers from the findings.

The similarity of their descriptors can be used to compare detected and reported features in images. In real-time applications, discovering image feature correspondences in two (or more) separate photos is a complicated subject, especially with high-resolution and huge images. It is important to balance between time efficiency and quality of found matches.

The brute-force descriptor matcher matches features using a brute-force technique. It uses distance calculations to compare the descriptor of one feature in the first image to the descriptors of all features in the second image. Then, in a resulting pair, the one that is closest is returned. The brute-force technique takes longer to run due to the massive number of comparisons, but it is extremely exact. Setting particular parameters can increase its performance and outlier elimination capability.

The most straightforward method for matching keypoints is to set a global threshold based on the Euclidean distance between descriptors. However, due to the large complexity of the feature space, this method has low accuracy because certain descriptors are significantly more exclusionary than others.

As a result, G.Lowe [9] considers not only the distance between a keypoint and the first most similar keypoint but also the distance between the keypoint and the second most similar keypoint; specifically, he uses the ratio between the distance between the candidate match and the distance between the second most similar feature point (i.e. the so-called 2NN test). This ratio must be less than a predetermined threshold to be considered a match (often not more than 0.6). Thus, in our case, we've considered it as 0.5. When a region is duplicated only once, this strategy works effectively, but not when it is copied multiple times.

## 8. Performance Evaluation and Testing

8.1 Discuss various test plans( Test Case No, Description, Input, Desired Output, Result of test case)

**Test case 1:** Copy move forgery done once

Input: Forged image of ducks swimming in a lake



Fig 17: test case 1

Desired Output: Highlight the area of the ducks which are copy and pasted below.

Output: Successfully detected and highlighted the forged area.

**Test case 2:** Copy move forgery done twice

Input: Forged image of ducks swimming with copy move forgery done twice in the same image.



Fig 18: Test case 2

Desired output: Highlight the area of the ducks which are copy and pasted below.

Output: Successfully detected and highlighted the forged area.

**Test case 3:**Copy move forgery scaled

Input: Forged image of a gun showcase. The copy move forgery part is scaled by 10%.



Fig 19: Test case 3

Desired output: Highlight the area of the gun which is copy and pasted above.

Output: Successfully detected and highlighted the forged area.

**Test case 4:** Copy move forgery scaled and rotated:

input: Forged image of car being forged. It is scaled and rotated.



Fig 20: Test case 4

Desired output: Highlight the area of the car which are copy and pasted below.

Output: Successfully detected and highlighted the forged area.

## 9. Result and Analysis

### 9.1 Explanation: how experiment has been performed

After successfully completing the code of SURF and SIFT, we took forged images from the internet to test our model(s). We made a GUI, giving the user an option to upload the image to be tested. After the user successfully uploads the image, they can choose either of the methods (SIFT/SURF) to test it. The selected method marks and highlights the forged regions in the uploaded test image. The user can then clearly see at which area is the image forged. This, will further help them resolve their queries and gain clarity about the reality of the picture.

Thus, we have successfully implemented SIFT and SURF techniques to detect Copy-Move Forgery. We have tested our project with 15 user input images, and have received a positive output. These images have different type of forgeries done in them. Our model performs accurately regardless of the scaling and rotation of the forged areas.



## 9.2 Discuss Results

Test case 1: Copy move forgery done once

Input: Forged image of ducks swimming in a lake



Fig 21: Result of test case 1

Test case 2: Copy move forgery done twice

Input: Forged image of ducks swimming in a lake

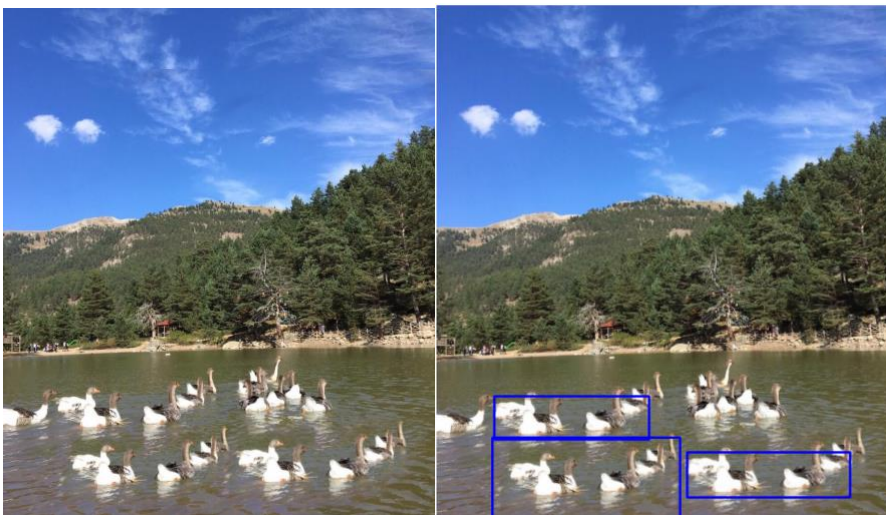


Fig 22: Result of test case 2

Test Case 3: Copy move forgery scaled

Input: Scaled Pistol images copy-pasted





Fig 23: Result of test case 3

Test Case 4: Copy move forgery scaled and rotated:

Input: Scaled and Roated Pistol images copy-pasted



Fig 24: Result of test case 4

## 10. Applications

The detection of image manipulation is very important because an image can be used as legal evidence, in forensics investigations, Health care, legal documents, Journalism to spread fear and in many other fields.

The pixel-based image forgery detection aims to verify the authenticity of digital images without any prior knowledge of the original image

## 11. Conclusion

Thus, we have successfully implemented SIFT and SURF techniques to detect Copy-Move Forgery. We have tested our project with 15 user input images, and have received a positive output. We've also implemented the RANSAC algorithm to remove outliers and hence increase the overall accuracy of our model. The given project successfully highlights the forged regions in different test cases and helps users with smooth detection of the same. The GUI created by us is user-friendly where in users can easily navigate through it and perform the functions as per their choice.

## **12. Future prospects of the project**

Currently our project demonstrates the detection of Copy-Move Forgery using SIFT and SURF techniques. We've implemented the same on different forged images and have received positive results. Future Scope of the project can be implementing various techniques to detect more types of forgeries such as splicing and image retouching. This will broaden the scope of our project. Moreover, we can work on creating a model that can detect any type of forgery from images and video clips. The creation of such a model will be even more complex and would require heavier machines and denser datasets.

### 13. References

- [1.] J. Fridrich, "Method for tamper detection in digital images", Proc. ACM Workshop on Multimedia and Security, pp. 19-23, 1999.
- [2.] I.T. Hsieh and Y.K. Wu, "Geometric invariant semi fragile image watermarking using real symmetric matrix", WSEAS Trans. of Signal Processing, vol. 2, no. 5, pp. 612-618, 2006.
- [3.] H. Farid, "image forgery detection - a survey", IEEE Signal Processing Magazine, vol. 5, pp. 16-25, 2009.
- [4.] G. Muhammad, M.H. Al-Hammadi, M. Hussain, and G. Bebis, "Image forgery detection using steerable pyramid transform and local binary pattern ", Machine Vision and Applications, DOI: 10.1007/s00138-013-0547-4, 2013.
- [5.] J. Wang, G. Liu, H. Li, Y. Dai, and Z. Wang, "Detection of image region duplication forgery using model with circle block", in Proc. Int. Conf. Multimedia Inf. Netw. Secur. (MINES), Nov. 2009, pp. 25–29.
- [6.] Mohammad Farukh Hasmi, Aaditya R. Hambarde, Avinash G. Keskara, "copy move forgery detection using DWT and SIFT features", 2013, Int. Conf. Intelligent systems design and applications.
- [7.] Chi-Man Pun, Xiao-Chen Yuan and Xiu-Li Bi, "Image forgery detection using Adaptive over segmentation and feature point matching", IEEE Trans. Inf. Forensics Security, vol. 10, Aug 2015.
- [8.] D. G. Lowe, "Object recognition from local scale- invariant features", in Proc. 7th IEEE Int. Conf. Comput. Vis., Sep. 1999, pp. 1150-1157.
- [9.] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60(2), 91–110. doi:10.1023/b:visi.0000029664.996

## 14. Appendices

### 14.1 Plagiarism Report from any open source/propriety source

Untitled document


endeavor to improve the identification phase for replicated image patches with highly uniform texture when prominent keypoints are not recovered using SIFT-like techniques. We'd like to improve the detection process by using higher recall rates and other keypoint-based approaches like SURF.

8 References

1. J. Fridrich, "Method for tamper detection in digital images", Proc. ACM Workshop on Multimedia and Security, pp. 19-23, 1999.
2. I.T. Hsieh and Y.K. Wu, "Geometric invariant semi fragile image watermarking using real symmetric matrix", WSEAS Trans. of Signal Processing, vol. 2, no. 5, pp. 612-618, 2006.
3. J. Fridrich, "Image forgery detection (survey)", IEEE Signal

Plagiarism

Back to all suggestions X



Looks like your text is 100% original.  
We found no matching text in our databases or on the Internet.

