**ABSTRACT**

The elevation of colour printing technology has increased the rate of fake currency note printing on a very large scale. It is the biggest problem faced by many countries including India. The effects that counterfeit money has on society include a reduction in the value of real money. There are machines present at banks and other commercial areas to check the authenticity of the currencies. But a common man does not have access to such systems and hence a need for software to detect fake currency arises, which can be used by common people. Verification of currency note is done by the concepts of image processing and segmentation. To remove such unwanted data in an image, number of the image pre-processing systems is required in order to better perception of the images. The way toward upgrading pixel intensity and image quality additionally are managed after pre handling. In this research, the pre-processing techniques like different types of noise and filters. The noises are speckle, poisson and Gaussian. Different filtering techniques can be adapted for noise declining to improve the visual quality as well as reorganization of images. The filters are Mean, Gaussain and Median filter are used to enhance the clarity of currency images. The purpose of Image segmentation is to partition an image into meaningful regions with respect to a particular application. Image segmentation is a method of separating the image from the background and read the contents. In this research through study has been done on most commonly used edge detection techniques such as Canny-edge detection, Laplacian and Gradient (sobel operator) comparatively reveal and analyze the distinctions from those processed images with those techniques use the peak signal-to-noise ratio (PSNR) for finding the best filter for removing noise from the image.

**CHAPTER-1**

**1. Introduction**

**1.1General Background**

Image processing is the most important part of the machine vision. Machine Vision is becoming more and more important these days because it is being used in manufacturing, inspection of parts, medical applications and robot guiding. Machine vision means making the machines to interact with the environment as human beings do in terms of seeing. But there is much work left to be done. Visual data are most complex and most useful sensory input for humans. Machine vision is concerned with the interpretation of similar visual data. Image Processing is the science of modifying and analyzing pictures. For analysis of images we need to find their edges first. Edges information can be used for segmentation of images or for locating their boundary in formation. Edge detection is important because edges give the compact description of the objects and objects can be reconstructed from the edge information.

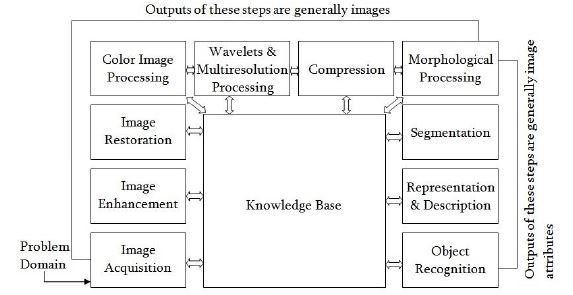
Vision begins with transformation of a flux of photons into a set of intensity

values at an array of sensors. The first step in visual information processing is to obtain a compact description of the raw intensity values. The primitive elements of the initial

description should ideally be complete in the sense of representing the full information contained in the image, and meaningful (that is, capturing significant properties of the three-dimensional surfaces around the viewer). Physical edges are among the most of objects since they correspond to object boundaries or to changes in surface orientation or material properties.

**BLOCK DIAGRAM OF IMAGE PROCESSING:**

Fundamental steps in digital image processing are shown in Fig. Image acquisition digitizes the image captured by camera. Image enhancement is the process of manipulating an image so that the results are more suitable for specific applications. Image restoration improves an appearance of an image, which tends to probabilities model of image degradation Morphological processes are the tools of extracting image components that are useful in the description and presentation of an image. Image segmentation is the most difficult ask in digital image processing which separates objects from the background. Representation makes the decision whether to represent data as boundary or as a complete region. Recognition is the process that assigns label to an object based on information provided by its descriptor.



**Block Diagram of Image Processing**

**1.2Motivation**

The main motivation behind development of this project was to make a system for easy and quick detection of genuine and fake currency notes. This is a Python based system for segmentation process applied for currency images. Since many years counterfeiting of paper currency challenges the financial system of every country in different sectors, India is also one of them. Modernization of the financial system is a milestone in protecting the economic prosperity, and maintaining social harmony. Automatic machines capable of recognizing banknotes are massively used in automatic dispensers of a number of different products, ranging from cigarettes to bus tickets, as well as in many automatic banking operations. The simplest way is to make use of the visible features of the paper currency, for example, the size and color of the paper currency. Our project can be a very good utility in banking systems and other field also. Fake notes in India in denominations of Rs.100, 500 and 2000 are being flooded into the system. Over the past few years, as results of the great technological advances in color printing, duplicating, and scanning, counterfeiting problems have become more and more serious. Automatic methods for paper currency detection become important in many applications such as automated teller machine and automated goods seller machines. This system is designed to recognize and verify the Indian paper currency. The approach consists of a number of steps including image a preprocessing and image segmentation images. This is a challenging issue to system designers. Every year RBI (Reserve bank of India) face the counterfeit currency notes or destroyed notes. Handling of large volume of counterfeit notes imposes additional problems. Therefore, involving machines (independently or as assistance to the human experts) makes notes detection process simpler and efficient.

**1.3 Objective**

The main objective of this project is fake currency detection using the image processing. Fake currency detection is a process of finding the forgery currency. After choose the image apply preprocessing. In pre-processing the image to be crop, smooth and adjust. Convert the image into gray color. After conversion apply the image segmentation. The Aim is to increasing the accuracy of determining if a note is genuine or counterfeit. There were many methods in existence, this method was designed to overcome the drawbacks of the previous methods. This method gives a faster and more accurate output when compared to the other techniques.

**1.4 Problem Statement**

The Existing System, doesn’t scanned the rear part of the currency note it only scans the front end part of the currency note. The existing system doesn’t give efficient results and it’s accuracy rate is low. Image processing and pattern recognition area is the base area for feature extraction. Existing system is a time consuming process for detection. There are some techniques which are specifically from the application area of Image processing and pattern recognition and using those techniques the researchers achieved the accuracy rate low. The main problem in devising an edge detector is that edges are a high frequency phenomenon and so is the noise in the image. To avoid detecting noise as edges a low pass filter or a large (in spatial extent) operator is used to average out the noise. But this operation also averages out the edges and is likely to form a single averaged edge from several neighboring edges. Consequently, this degrades the position accuracy of the detected edges.

**1.5 Overview of Working Process**

Convert RGB to Gray scale Image

Find Speckle, Poisson and Gaussian Noises

Mean Filter

Gaussian Filter

Median Filter

Segmentation

Gradient

Laplacian

Canny-Edge

**1.6 Types of Image Noise**

**a. Speckle Noise**

Speckle noise is one kind of granular noise and the picture quality has been degraded by this speckle noise. The images which are acquired from medical are spoiled by the speckle noise. Generally, speckle noise expands the mean gray near of a native area and causing difficulties in medical because of coherent processing of backscattered signal.

**b. Poisson Noise**

Poisson noise also called photon noise. It is happening due to some statistical variation in the assessment. The uncertainty associated with this type of noise is the measurement of light and independent of photons . The expected

magnitude of the signal is signal dependent and independent of low light conditions.

Reducing noise from the digital image is a test for the specialists in the preprocessing phase. There are different filtering methods that are accessible for various sorts of noise reductions.

**c. Gaussian Noise**

Gaussian noise is likewise called enhancer noise or random variation impulsive noise. Gaussian noise is created because of (a) electronic circuit noise, (b) sensor noise because of high temperature, (c) sensor noise because of poor brightening. It is a sort of measurable noise where the sufficiency of the noise takes after Gaussian dissemination. Gaussian noise emerges as probability density function of the regular distribution.

**1.7 Types of Filters**

**a. Mean Filter**

Mean filtering simplest way to decrease the amount of intensity deviation between one and next pixel for smoothing image.

At this time, g is the ruined image, r and c are the row and column co-ordinates correspondingly within a window size of m×n besides the filtered image. However geometric mean filter is a variation of the arithmetic mean filter.

**b. Gaussian Filter**

Gaussian filter is a linear smoothing filter, where the weights chosen for the smoothing purpose according to the outline of the function of Gaussian.

A Gaussian Filter is a low pass filter used for **reducing noise (high frequency components) and blurring regions of an image**.

**c. Median Filter**

Order-statistics filter also known as Median filter, which exchanges the estimation of a pixel by the middle of the gray levels in the region of that pixel. The median is a rank command statistic and in a intelligence the main stream of the pixel values included determines the result.

**1.8 Types of Segmentation Filters**

**a. Canny-Edge**

Canny edge detector is one of the most commonly used image processing tools. The Gaussian smoothing in the canny edge detector fulfills two purposes: first it can be used to control the amount of detail that appears in the edge image and second, it can be used to suppress noise.

It is a better method because it extracts the features in an image without disturbing its features. There are certain criteria to improve current methods of edge detection. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed. The second criterion is that the edge points be well localized i.e. the distance between the edge pixels as found by the detector and the actual edge should be minimum. A third criterion is to have only one response to a single edge.

**b. Laplacian**

In laplacian based methods, edges are recognized by searching zero crossings in a 2nd derivative of the image. A pre-processing step is taken that is smoothing usually Guassian smoothing of image. In smoothing, firstly convolution is performed between image and Guassian filter to minimize noise. Thereafter isolated noise pixels and small structures are filtered out. Laplacian of gaussian is also known as Marr-Hildreth Edge Detector.

The Laplacian edge detector uses only one mask. It can compute second order derivatives in a one pass.

**c. Gradient**

The edge is the area where image grey value changes abruptly, using gradient operator edges are detected by perfoming first order derivative. It detect edges by computing the magnitude of gradient, and then going for local directional maxima of it by using an estimation of the local orientation of the edges, normally the gradient’s direction. For an image function f(x, y), the derivative of ‘f’ at coordinates (x, y) is denoted as the two directional column vector.

Sobel Operator: It is 3x3 convolution kernels. One kernel is simply the other rotated by 900. It is a row-edge detector.

The kernel can be applied separately to input image for obtaining gradient component in each orientation i.e. GX and GY.

**CHAPTER 2**

**LITERATURE SURVEY**

1. **D. Alekhya, G. Devisuryaprabha, and G. Venkatadurgarao, “Fake Cur- rency Detection Using Image Processing and Other Standard Methods,” Int. J. Res. Comput. Commun. Technol. ISSN, vol. 3, no.1, pp. 2278–5841, 2020.**

**Description:**

In the last eight years more than 3.53 lakh cases of counterfeit currency detection in India’s banking channels is heighten according to latest government reports. The practice of counterfeiting became more refined with the arrival of paper currency. The Indian Government has taken a astonishing stride of demonetizing 500 and 1000 Rs. notes. Prime Minister Shree. Narendra Modi stated that one of the recognition for this policy was to counter the climbing menace of counterfeit Indian Currency notes. However, the Indian banks acknowledged an all-time peak amount of fake currency and also noticed an over 480% increment in doubtful transactions after demonetization, a first ever report on questioning credits ended in the wake of 2016 notes ban has discovered. The Reserve Bank of India (RBI) is the only one which has the singular authority to issue bank notes in India.. **Disadvantages :**

* This application, are able to see the missing parameters which the fake note doesn’t have as compared to the original notes.

**Advantages:**

* The work will surely be very useful for minimizing the counterfeit currency.

**2 P. P. MDeborah and P. CSoniya PrathapME, “Detection of Fake currency using Image Processing,” IJISET -International J. Innov. Sci. Eng. Technol., vol. 1, no. 10, pp. 151–157, 2021.**

**Description:**

Fake currency detection is a serious issue worldwide, affecting the economy of almost every country including India. The use of counterfeit currency is one of the major issues faced throughout the world nowadays. The counterfeiters are becoming harder to track down because of their use of highly advanced technology. One of the most effective methods to stop counterfeiting can be the use of counterfeit detection software that is easily available and is efficient. Our project will recognize Indian currency notes using a real-time image obtained from a webcam. The background of our topic is image processing technology and applying it for the purpose of verifying valid currency notes. The software will detect fake currency by extracting features of notes. The success rate of this software can be measured in terms of accuracy and speed. So our aim is to work on those parameters which will be impossible to implement on counterfeit notes so we started working on minutiae parameters which will be enough to differentiate between fake and original notes.

**Disadvantages:**

* For fake notes detection, need to focus on the key frames and joint information, and too much redundant will lead to a decrease in fake notes recognition.

**Advantages:**

* Skeleton based fake notes detection has attracted extensive attention due to the robustness of the fake notes skeleton data in the field of computer vision.

**3**. **M. R. Pujar. ”Indian Currency Recognition and Verification using Image Processing”, International Journal of Advance Research, Ideas and Innovations in Technology vol. 3, pp. 175–180, 2018.**

**Description:**

Indian is a developing country, Production, and printing of Fake notes of Rs.100, 500 and 1000 were already there but after the demonetization, the counterfeit notes of new Rs.50,200,500,2000 have also come to the light in very short time and which effects the country’s economic growth. From last few years due to technological advancement in color printing, duplicating, and scanning, counterfeiting problems are coming into the picture. In this article, recognition and verification of paper currency with the help of digital image processing techniques is described. The characteristics extraction is performed on the image of the currency and it is compared with the characteristics of the genuine currency. The currency will be recognized and verified by using image processing techniques.

**Disadvantages:**

* It is not helpful if the note is dirty embedded in the Indian economy that even bank branches and ATMs are disbursing counterfeit currency.

**Advantages:**

* Identifying and extracting visible and invisible features of currency notes.
* The best way is to use the visible features of the note

### 4. T. S. Kim and A. Reiter, “Interpretable 3D fake notes fake notes analysis with temporal convolutional networks,” in Proc. IEEEConf. Comput. Vis.Pattern Recognition. Workshops (CVPRW), Jul. 2017, pp. 1623-1631.

### Description:

The discriminative power of modern deep learning models for 3D fake notes fake notes recognition is growing ever so potent. In conjunction with the recent resurgence of 3D fake notes fake notes representation with 3D skeletons, the quality and the pace of recent progress have been significant. However, the inner workings of state- of-the-art learning-based methods in 3D fake notes fake notes recognition still remain mostly black-box. In this work, we propose to use a new class of models known as Temporal Convolutional Neural Networks (TCN) for 3D fake notes recognition. Compared to popular LSTM-based Recurrent Neural Network models, given interpretable input such as 3D skeletons, TCN provides us a way to explicitly learn readily interpretable spatio-temporal representations for 3D fake notes fake notes recognition.

**Disadvantages:**

• HSOM has not been tested on the spatial temporal input space and it has been used to classify documents.

**Advantages:**

• The hybrid model show excellent performance on activity recognition of one person activity.

**5. Z. Yang, Y. Li, J. Yang, and J. Luo, “Fake notes recognition with spatiofi temporal visual attention on skeleton image sequences”, IEEE Trans. CircuitsSyst. Video Technol., vol. 29, no. 8, pp. 2405-2415, Aug. 2019.**

**Description:**

Fake notes recognition with 3D skeleton sequences is becoming popular due to its speed and robustness. The recently proposed Convolutional Neural Networks (CNN) based methods have shown good performance in learning spatio-temporal representations for skeleton sequences. Despite the good recognition accuracy achieved by previous CNN based methods, there exist two problems that potentially limit the performance. First, previous skeleton representations are generated by chaining joints with a fixed order. The corresponding semantic meaning is unclear and the structural information among the joints is lost. Second, previous models do not have an ability to focus on informative joints. The attention mechanism is important for skeleton-based fake notes recognition because there exist spatio-temporal key stages while the joint predictions can be inaccurate. To solve these two problems, we propose a novel CNN based method for skeleton-based fake notes recognition. We first redesign the skeleton representations with a depth-first tree traversal order, which enhances the semantic meaning of skeleton images and better preserves the associated structural information.

**Disadvantages:**

• One fixed graph structure may not be optimal for all the samples of different fake notes classes.

**Advantages:**

• The structure of GCN is hierarchical where different layers contain multi level semantic information.

**6. S. Yan, Y. Xiong, and D. Lin, “Spatial temporal graph convolutional networks for skeleton-based fake notes recognition,’’ in Proc. 32 nd AAAI Conf. Artif. Intell., 2018, pp. 11-20.**

**Description:**

Dynamics of fake notes body skeletons convey significant information for fake notes fake notes recognition. Conventional approaches for modelling skeletons usually rely on hand-crafted parts or traversal rules, thus resulting in limited expressive power and difficulties of generalization. In this work, we propose a novel model of dynamic skeletons called Spatial-Temporal Graph Convolutional Networks (ST-GCN), which moves beyond the limitations of previous methods by automatically learning both the spatial and temporal patterns from data. This formulation not only leads to greater expressive power but also stronger generalization capability. On two large datasets, Kinetics and NTU-RGBD, it achieves substantial improvements over mainstream methods.

**Disadvantages:**

• HSOM has not been tested on the spatial temporal input space and it has been used to classify documents.

**Advantages:**

• The hybrid model show excellent performance on activity recognition of one person activity.

**7. Gajanand Gupta, “Algorithm for Image Processing Using Improved Median Filter and Comparison of Mean and Improved Median Filter” (IJSCE) ISSN: 2231-2307, Volume-1, Issue-5, November 2021.**

**Description:**

Noise reduction in medical images is a perplexing undertaking for the researchers in digital image processing. Noise generates maximum critical disturbances as well as touches the medical images quality, ultrasound images in the field of biomedical imaging. The image is normally considered as gathering of data and existence of noises degradation the image quality. It ought to be vital to re-establish the original image noises for accomplishing maximum data from images. Medical images are debased through noise through its transmission and procurement. Image with noise reduce the image contrast and resolution, thereby decreasing the diagnostic values of the medical image. This paper mainly focuses on Gaussian noise, Pepper noise, Uniform noise, Salt and Speckle noise. Different filtering techniques can be adapted for noise declining to improve the visual quality as well as reorganization of images. Here four types of noises have been undertaken and applied on medical images.

**Disadvantages:**

* Very sensitive to noise.
* Less accurate in finding Orientation of edges.

**Advantages:**

* Orientation is possible.
* Fixed characteristic in all Direction.

**8. Y.Ramadevi, T.Sridevi, B.Poornima, B.Kalyani “Segmentation and object recognition using edge detection techniques” International Journal of Computer Science & Information Technology (IJCSIT), Vol 2, No 6, December 2020**

**Description:**

Image segmentation is to partition an image into meaningful regions with respect to a particular application. Object recognition is the task of finding a given object in an image or video sequence. In this paper, interaction between image segmentation (using different edge detection methods) and object recognition are discussed. Edge detection methods such as Sobel, Prewitt, Roberts, Canny, Laplacian of Guassian (LoG) are used for segmenting the image. Expectation-Maximization (EM) algorithm, OSTU and Genetic algorithms were used to demonstrate the synergy between the segmented images and object recognition.

**Disadvantages:**

* Discontinuity in edges.
* Sensitive to noise.
* Can’t meet real time Requirement.

**Advantages:**

* Easy to implement.
* Orientation is possible.

**9. Sunanda Gupta, Charu Gupta & S.K.Chakarvarty “Image edge detection :A review” international journal of advance reaserach in computer science and technology (IJARCET) Volume 2, Issue 7, July 2021.**

**Description:**

Edges are the set of curved line segments where brightness level of image changes sharply. It is one of the most important information of an image which can helps to detect object boundary, its relative position within target area and many other useful information. In edge detection process, edges are retrieved from an image by spotting high intensity variations of the pixels. Edge detection of an image minimizes the amount of processed data effectively and discards information that is less important, keeping the important structural properties of an image. This paper presents a different approach to apply Gradient and LoG operator to get more continuous edges than the conventional one using MATLAB. Their results are compared using peak signal to noise ratio (PSNR). Two images in rainy weather are taken by my camera for case study. It can be used in many applications such as in object tracking, in data compression, in image analysis and medical imaging.

**Disadvantages:**

* Complex computation.
* Low operational speed.

**Advantages:**

* Better performance in case of Noise.
* Reduce computation time Effectively.

**10.** **G.T.Shrivakshan & Dr.C.Chandrasekar, “A comparison of various edge detection techniques used in image processing” IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 1, September 2012.**

**Description:**

Authentication and Identification is one of the key features of Biometric Application. In such applications before generating the templates, a feature is extracted from the input image captured by a sensor after pre-processing. In pre-processing steps, the goal is to enhance the visual appearance of the image by noise removal, dilation, erosion, segmentation etc. In feature extraction, the edge is detected where there is an abrupt change in the intensity values of the image. This paper is aimed to analyse various edge detection techniques like Prewitt, Sobel, Roberts, Canny, LoG, Zero crossing etc. and proposing the best suitable method of edge detection for biometric application. The comparison of biometric image edge detection is based on the comparison parameter Mean Square Error (MSE), Root Mean Square Error (RMSE), Peek Signal to Noise Ratio (PSNR) using MATLAB software.

**Disadvantages:**

* Complex and time consuming computations.
* False zero crossing.
* It is difficult to give a generic threshold that works well on all image.

**Advantages:**

* Better detection in noise conditions.
* Retains texture of original image.
* Less computation time.

**CHAPTER 3**

**METHODOLOGY**

**3.1 Introduction**

In this method characteristics of paper currencies are employed that are used by people for differentiating different banknote denominations. Basically, at first instance, people may not pay attention to the details and exact characteristics of banknotes for their recognition, rather they consider the common characteristics of banknotes such as the size, the background color (the basic color), and texture present on the banknotes. In this method, these characteristics will be used to differentiate between different banknote denominations. The characteristics extraction is performed on the image of the currency and it is compared with the characteristics of the genuine currency. The Canny operator with gradient magnitude is used for characteristic extraction. The currency will be verified by using image processing techniques. The use of serial number extraction is if any counterfeit note is encountered we can immediately send the report about that counterfeit note. Image segmentation is the process of partitioning a digital image into multiple image segments, also known as image regions or image objects (sets of pixels). By dividing an image into segments, the process only the important segments of the image instead of processing the entire image. The purpose of Image segmentation is to partition an image into meaningful regions with respect to a particular application. Image segmentation is a method of separating the image from the background and read the contents. In this research through study has been done on most commonly used edge detection techniques such as Canny-edge detection, Laplacian and Gradient (sobel operator).

**3.2 Methodology of Pre-processing and Segmentation**

**Input Image**

**RGB to Gray scale Conversion**

**Images Resize**

**Image Pre-Processing(Noise Removal and Filtering)**

**Segmentation**

**Output Image**

**3.3 Module Description**

**3.3.1 Pre-Processing**

The aim of preprocessing is an improvement of an image data that suppresses unwanted distortions or enhances some image features important for further processing. The pre preparing associated with transformation, picture resize, noise removing and improves the quality and produces an image where in details can be perceived precisely. To remove such unwanted data in an image, number of the image pre-processing systems is required in order to better perception of the images. The way toward upgrading pixel intensity and image quality additionally are managed after pre handling. In this research, the pre-processing techniques like different types of noise and filters. The noises are speckle, poisson and Gaussian. Different filtering techniques can be adapted for noise declining to improve the visual quality as well as reorganization of images. The filters are Mean, Gaussain and Median filter are used to enhance the clarity of currency images.

**i. Grayscale Conversion:**

The filtering strategies like Gaussian and median channel are balanced for noise removal method. it is easy to remove the noise by using conversion of colours images to gray scale images. The image acquired is in RGB color. It is converted into gray scale because it carries only the intensity information which is easy to process instead of processing three components R (Red), G(Green), B(Blue).

**ii.Image resizes:**

Picture resizing is a critical job in picture taking care of strategy, to create and decrease the given picture size in pixel position. picture contribution can be divided into two distinct ways, they are picture down-testing and up-examining which is important while resizing the data for organizing either the specific correspondence channel or the output display. While it is more efficient to transmit low resolution forms to the client an estimation of the first high resolution might be required for introducing the last visual data. An exact resizing of picture data is a fundamental advance in numerous applications, extending from a few buyer things to essential limits inside the clinical security and defence segments. The speed of resizing can be resolved with the usage of strategy experiences the way that the subsequent picture frequently contains block artefacts, which are not outwardly noticeable but rather regularly additionally can definitely contrarily influence, error calculations wont to compare methods

**iii. Noise Removal:**

Image noise is described considering the way that the irregular variety of brightness or shading data in pictures conveyed by currency devices or scanners. Picture noise is commonly seen as a undesirable result during picture obtaining. Noise is generally characterized because the uncertainity during a signal. There are numerous explanations behind these variances. Each currency picture contain some visual noise. The presence of noise gives a picture a mottled, grainy, completed, or covered appearance. A couple of sorts of noise exist in currency pictures.

**A. Speckle Noise:**

Speckle noise is one kind of granular noise and the picture quality has been degraded by this speckle noise. The images which are acquired from medical are spoiled by the speckle noise. Generally, speckle noise expands the mean gray near of a native area and causing difficulties in medical because of coherent processing of backscattered signal. Speckle noise Pulse generator picture standard is mostly reduced due to speckle noise. Speckle noise is typical marvel in all intelligent imaging frameworks like a laser, acoustic and SAR symbolism.

**B. Poisson Noise:**

Poisson noise also called photon noise. It is happening due to some statisticalvariation in the assessment. The uncertainty associated with this type of noise is the measurement of light and independent of photons . The expected magnitude of the signal is signal dependent and independent of low light conditions. Reducing noise from the digital image is a test for the specialists in the pre-processing phase. There are different filtering methods that are accessible for various sorts of noise reductions.

**C. Gaussian Noise:**

Gaussian noise is measurable noise that includes a likelihood thickness work (PDF) of the ordinary distribution, also referred to as regular assignment. In Gaussian noise, each pixel inside the picture will be changed from its unique value by a (normally) bit. It is also called as electronic noise because it arises in amplifiers or detectors.

Source: thermal vibration of atoms and discrete nature of radiation of warm objects.



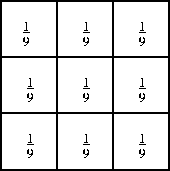
The side image is a bell shaped probability distribution function which have mean 0 and standard deviation (sigma) 1.

**iv. Filtering Methods:**

Filtering technique for improving a picture , in filtering are essentially wont to cover either the high frequencies inside the image, for instance smoothing the image, or the low frequencies, i.e., improving or perceiving edges inside the image. For instance, you will filter a picture to pressure certain feature or evacuate different highlights. Number of techniques are available and thusly as well as can be expected depends on the picture and the manner in which it will be utilized. Picture filtering is important for a few applications, including smoothing, sharpening, expelling noise, and edge acknowledgment. filtering methods Input pictures are influenced by various noise either Gaussian channel, Median channel, etc. so noise picture process is managed to improve the image quality utilizing the filtering strategy respectively. The detail of each noise described below.

**A. Mean Filter:**

The simplest of these algorithms is the Mean Filter. The Mean Filter is a linear filter which uses a mask over each pixel in the signal. Each of the components of the pixels which fall under the mask are averaged together to form a single pixel. This new pixel is then used to replace the pixel in the signal studied. The Mean Filter is poor at maintaining edges within the image. It is often used to https://homepages.inf.ed.ac.uk/rbf/HIPR2/mote.gifreduce noise in images.

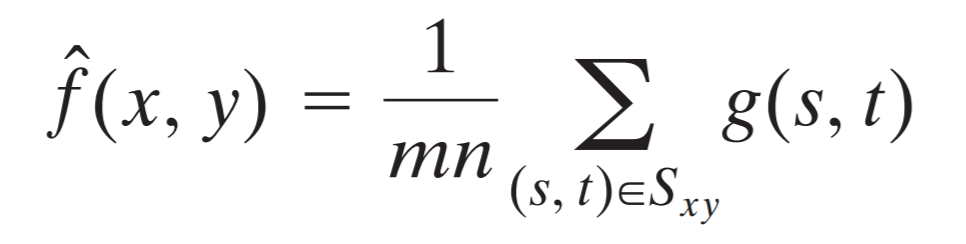


3×3 averaging kernel often used in mean filtering

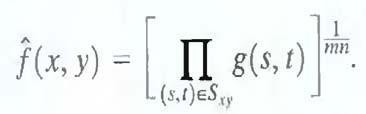
Computing the straightforward convolution of an image with this kernel carries out the mean filtering process. The idea of mean filtering is simply to replace each pixel value in an image with the mean (`average') value of its neighbors, including itself.

An arithmetic mean filter operation on an image removes short tailed noise such as uniform and Gaussian type noise from the image at the cost of blurring the image. The arithmetic mean filter is defined as the average of all pixels within a local region of an image.

Mean filtering simplest way to decrease the amount of intensity deviation between one and next pixel for smoothing image. Arithmetic mean filter can be expressed by,



At this time, g is the ruined image, r and c are the row and column co-ordinates correspondingly within a window size of m×n besides the filtered image is f’. However geometric mean filter is a variation of the arithmetic mean filter, which calculated appearance can be known as follows,



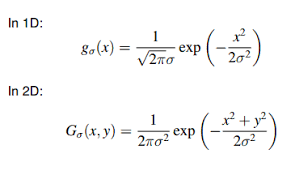
This has the effect of eliminating pixel values which are unrepresentative of their surroundings. Mean filtering is usually thought of as a https://homepages.inf.ed.ac.uk/rbf/HIPR2/mote.gifconvolution filter.

Like other convolutions it is based around a kernel, which represents the shape and size of the neighborhood to be sampled when calculating the mean.

**Algorithm:** **Step 1:** Place a window over element. **Step2:** Take an average — sum up elements and divide the sum by the number of elements.

**B. Gaussian Filter:**

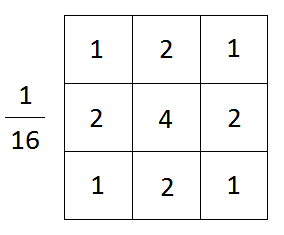
Gaussian filter is a linear smoothing filter, where the weights chosen for the smoothing purpose according to the outline of the function of Gaussian. Gaussian filter in the nonstop space and can be defined by the resulting equation,



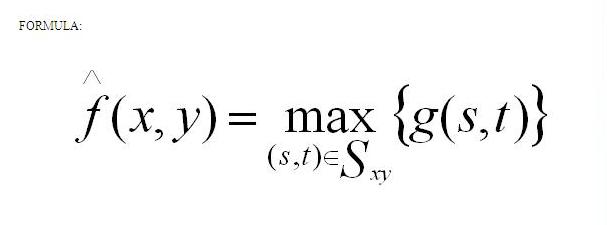
A Gaussian Filter is a low pass filter used for **reducing noise (high frequency components) and blurring regions of an image**. The kernel is not hard towards drastic color changed (edges) due to it the pixels towards the center of the kernel having more weightage towards the final value then the periphery. A Gaussian Filter could be considered as an approximation of the Gaussian Function (mathematics).

A further way to compute a Gaussian smoothing with a large standard deviation is to convolve an image several times with a smaller Gaussian. While this is computationally complex, it can have applicability if the processing is carried out using a hardware pipeline.

Using the above function a gaussian kernel of any size can be calculated, by providing it with appropriate values. A 3×3 Gaussian Kernel Approximation(two dimensional) with Standard Deviation = 1, appears as follows



**C. Median Filter:**

The median filter is the filtering technique used for noise removal from images and signals. Median filter is very crucial in the image processing field as it is well known for the **preservation of edges during noise removal.** Order-statistics filter also known as Median filter, which exchanges the estimation of a pixel by the middle of the gray levels in the region of that pixel. The median is a rank command statistic and in a intelligence the main stream of the pixel values included determines the result. The expression of Median filter,  The first estimation of the pixel is incorporated into the calculation of the median. Median filters are extremely standard for assured sorts of arbitrary noise. They give astonishing noise diminish capacities, with stunningly less clouding than a linear smoothing filter of comparative size. The median is figured by first sorting all the pixel values from the window in numerical order. A while later supplanting the pixel being considered with the inside (middle) pixel value. Median filters are viable for the bipolar and unipolar impulse noise. Median filters are mainly reasonable within the sight of both unipolar and bipolar impulse noise.

**Algorithm for the improved median filter**

To remove salt and pepper noise from the corrupted image the below described algorithm is used.

**Step 1**: A two dimensional window (denoted by 3×3 W) of size 3x3 is selected and centered around the processed pixel p(x, y) in the corrupted image.

**Step 2**: Sort the pixels in the selected window according to the ascending order and find the median pixel value denoted by Pmed), maximum pixel value (Pmax) and minimum pixel value (Pmin) of the sorted vector V0. Now the first and last elements of the vector V0 is the Pmin and Pmax respectively and the middle element of the vector is the Pmed.

**Step 3**: If the processed pixel is within the range Pmin < P(x, y) < Pmax , Pmin > 0 and Pmax < 255,it is classified as uncorrupted pixel and it is left unchanged. Otherwise p(x, y) is classified as corrupted pixel.

**Step 4:** If p(x, y) is corrupted pixel, then we have the following two cases:

**Case 1:** If Pmin < Pmed < Pmax and 0 < Pmed < 255, replace the corrupted pixel p(x, y) with Pmed.

**Case 2:** If the condition in case 1 is not satisfied then Pmed is a noisy pixel. In this case compute the difference between each pair of adjacent pixel across the sorted vector V0 and obtain the difference vector VD .Then find the maximum difference in the VD and mark its corresponding pixel in the V0 to the processed pixel.

**Step 5:** Step 1 to step 4 are repeated until the processing is completed for the entire image.

**3.3.2 Segmentation:**

Image segmentation is the process of partitioning a digital image into multiple image segments, also known as image regions or image objects (sets of pixels). By dividing an image into segments, the process only the important segments of the image instead of processing the entire image. The purpose of Image segmentation is to partition an image into meaningful regions with respect to a particular application. Image segmentation is a method of separating the image from the background and read the contents. In this research through study has been done on most commonly used edge detection techniques such as Canny-edge detection, Laplacian and Gradient (sobel operator). Essentially, in image partitions are different objects which have the same texture or color. The image segmentation results are a set of regions that cover the entire image together and a set of contours extracted from the image. All of the pixels in a region are similar with respect to some characteristics such as color, intensity, or texture. Adjacent regions are considerably different with respect to the same individuality. The different approaches are,

1. by finding boundaries between regions based on discontinuities in intensity levels,
2. thresholds based on the distribution of pixel properties, such as intensity values, and
3. based on finding the regions directly.

Thus the choice of image segmentation technique is depends on the problem being considered. **Steps in Edge detection:**

**A.** **Filtration:** Every image is associated with some intensity values, random change in these values can result in noise. Some common noise is: salt and pepper noise, impulse noise etc. Noise can result in difficulties in effective edge detection; hence image has to be filtered in order to reduce the noise content that leads to loss of edge strength . It is also termed as Smoothening.

**B. Enhancement:** Improving the quality of image is termed as enhancement. It aims to produce an image which is better and more suitable than original. A filter is applied in order to enhance the quality of edge in image.

**C. Detection:** Several methods are adopted to determine which points are edge points and which a edge pixels should be discarded as noise.

Edge detection is the problem of fundamental importance in image analysis. Edge detection techniques are generally used for finding discontinuities in gray level images. To detect consequential discontinuities in the gray level image is the important common approach in edge detection. Image segmentation methods for detecting discontinuities are boundary based methods.

**A. Canny-Edge Detection:**

The Canny edge detection technique is one of the standard edge detection techniques. It was first created by John Canny for his Master’s thesis at MIT in 1983, and still outperforms many of the newer algorithms that have been developed. To find edges by separating noise from the image before find edges of image the Canny is a very important method. Canny method is a better method without disturbing the features of the edges in the image. It is a better method because it extracts the features in an image without disturbing its features. There are certain criteria to improve current methods of edge detection. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed. The second criterion is that the edge points be well localized i.e. the distance between the edge pixels as found by the detector and the actual edge should be minimum. A third criterion is to have only one response to a single edge.

**Importance of Canny**

Despite of number of edge detection techniques available canny algorithm is considered because it contains a number of adjustable parameters which can affect the computation time and effectiveness of the algorithm.

a) The size of the Gaussian filter: The smoothing filter used in the first stage directly affects the results of the detection of small, sharp lines. A larger filter causes more blurring, smearing out the value of an given pixel over a larger area of image.

b) The use of two thresholds with hysteresis allows more flexibility than in a single-threshold. A threshold set too high can miss important information. On the other hand, a threshold set too low will falsely identify irrelevant information (such as noise) as important.

The edge detection in this technique is optimized with regard to the following criteria.

a) Maximizing the signal-to-noise ratio of the gradient.

b) Edge localization for ensuring the accuracy of edge.

c) Minimizing multiple responses to a single edge.

**The algorithmic steps are as follows:**

**Step 1:** Convolve image f(r, c) with a Gaussian function to get smooth image

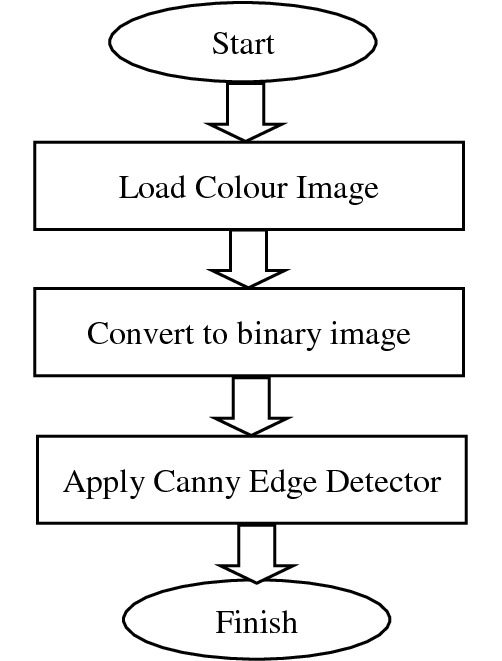
f^(r, c). f^(r, c)=f(r,c)\*G(r,c,6)

**Step 2:** Apply first difference gradient operator to compute edge strength then edge magnitude and direction are obtain as before.

**Step 3:** Apply non-maximal or critical suppression to the gradient magnitude.

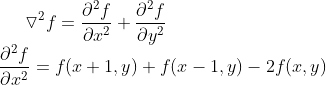
**Step 4:** Apply threshold to the non-maximal suppression image.

**Flow chart**



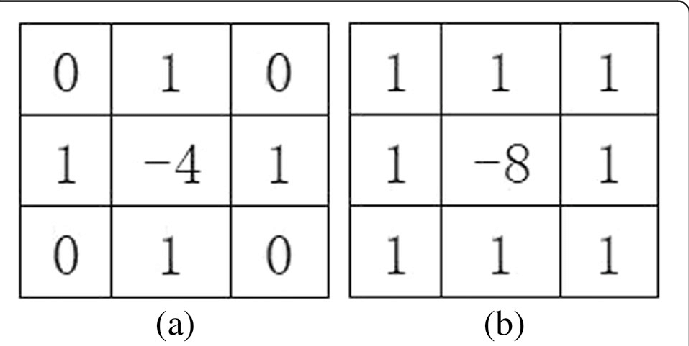
**B. Laplacian**

The Laplacian edge detection algorithm usually referred to Log detection algorithm. Also known as Laplacian filtering operator, it is a direction-free, isotropic operator directly take the 2nd order spatial derivative of an image. The distinction of this method is that the log operator only focuses on the rapid discontinuity of the pixel intensity within a certain area. That could run into trouble if images are raw and high-frequency noise hasn’t been smoothed before this step. As a result, a smooth process is usually pre-requisite for this algorithm. Typically, Log is taking a grayscale image no matter what the type of the gray image is, and produce another one in grayscale. It first smoothes the image and then computes the Laplacian. This yields in double edge image; hence for finding the edge the zero crossing between the double edges is taken. The Laplacian of an image with the pixel intensity value L(x,y) is given by:



In laplacian based methods, edges are recognized by searching zero crossings in a 2nd derivative of the image. A pre-processing step is taken that is smoothing usually Guassian smoothing of image. In smoothing, firstly convolution is performed between image and Guassian filter to minimize noise. There after isolated noise pixels and small structures are filtered out.

The commonly used discrete approximations to Laplacian filter are:

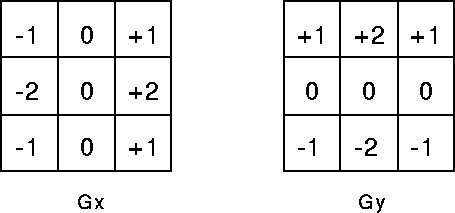


The Gaussian filtering is combined with Laplacian to break down the image where the intensity varies to detect the edges effectively. **Steps:**

* Read the image
* If the image is colored then convert it into RGB format.
* Define the Laplacian filter.
* Convolve the image with the filter.
* Display the binary edge-detected image.

**C. Gradient:**

Sobel Operator: It is 3x3 convolution kernels. One kernel is simply the other rotated by 900. It is a row edge detector.



GX and GY are the common masks used in Sobel Operator.

This figure shows the masks used by Sobel operator. The kernel can be applied separately to input image for obtaining gradient component in each orientation i.e. GX and GY.

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these *Gx* and *Gy*). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

Eqn:eqnrob1

And its approximation is done by:

Eqn:eqnrob2

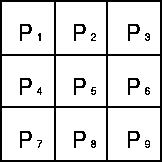
which is much faster to compute.

The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

Eqn:eqnsob3

In this case, orientation 0 is taken to mean that the direction of maximum contrast from black to white runs from left to right on the image, and other angles are measured anti-clockwise from this.

Often, this absolute magnitude is the only output the user sees --- the two components of the gradient are conveniently computed and added in a single pass over the input image using the pseudo-convolution operator shown in Fig.



Pseudo-convolution kernels used to quickly compute approximate gradient magnitude Using this kernel the approximate magnitude is given by:

Eqn:eqnsob1

**The algorithm works with two kernels** 1. A kernel to approximate intensity change in the x-direction (horizontal). 2. A kernel to approximate intensity change at a pixel in the y-direction (vertical).

**CHAPTER 4**

**IMPLEMENTATION DETAILS**

**4.1 HARDWARE REQUIREMENTS**

• Processor: Intel i3

• Hard disk: 250 GB

• RAM: 4 GB

**4.2 SOFTWARE REQUIREMENTS**

• IDE: Jupyter

• Front End: Anaconda Python

• Back End: Dataset

**4.3 INTRODUCTION TO SOFTWARE**

**4.3.1 Introduction to Python**

Python is an interpreted, high-level and general-purpose programming language. Python’s design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a “batteries included” language due to its comprehensive standard library. Python was created in the late 1980s, and first released in 1991, by Guido van Rossum as a successor to the ABC programming language. Python 2.0, released in 2000, introduced new features, such as list comprehensions, and a garbage collection system with reference counting, and was discontinued with version 2.7 in 2020. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible and much Python 2 code does not run unmodified on Python 3.

Python interpreters are supported for mainstream operating systems and available for a few more (and in the past supported many more). A global community of programmers develops and maintains CPython, a free and open-source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development. It currently ties with Java as the second most popular programming language in the world.

**Features**

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by meta programming and meta objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names

during program execution.

Python’s design offers some support for functional programming in the Lisp tradition. It has filter, map, and reduce functions; list comprehensions, dictionaries, sets, and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

The language’s core philosophy is summarized in the document The Zen of

Python (PEP 20), which includes aphorisms such as:

• Beautiful is better than ugly.

• Explicit is better than implicit.

• Simple is better than complex.

• Complex is better than complicated.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum’s vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl “there is more than one way to do it” motto, Python embraces a “there should be one—and preferably only one—obvious way to do it” design philosophy. Alex Martelli, a Fellow at the Python Software Foundation and Python book author, writes that “To describe something as ‘clever’ is not considered a compliment in the Python culture”.

Python’s developers strive to avoid premature optimization, and reject patches to non-critical parts of the CPython reference implementation that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter. An important goal of Python’s developers is keeping it fun to use. This is reflected in the language’s name — a tribute to the British comedy group Monty Python — and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard foo and bar.

A common neologism in the Python community is pythonic, which can have a wide range of meanings related to program style. To say that code is pythonic is to say that it uses Python idioms well, that it is natural or shows fluency in the language, that it conforms with Python’s minimalist philosophy and emphasis on readability. In contrast, code that is difficult to understand or reads like a rough transcription from another programming language is called unpythonic. Users and admirers of Python, especially those considered knowledgeable or experienced, are often referred to as Pythonistas.

**OpenCV**

Open CV (Open-Source Computer Vision Library) is a open source computer vision software library for the purpose of machine learning. Open CV was developed to serve the purpose of computer vision applications and to stimulate the usage of machine perception in the commercially viable products. Open CV is a BSD-licensed product which is easy for the utilization and modification of the code. The library contains more than 2500 advanced algorithms including an extensive set of both typical and state-of-the-art computer vision and machine learning algorithms. These algorithms can be employed for the detection and recognition of faces, identification of objects, extra fake notes of 3 D models of objects, production of 3 D point clouds from stereo cameras, stitching images together for production of a high resolution image of an entire scene, finding similar images from an image database, removing red eyes from images taken using flash, following ye movements, recognition of scenery and establishing markers to overlay it with intensified reality etc. It includes C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. Open CV mainly involves real-time vision applications taking advantage of MMX and SSE instructions when available. A full-featured CUDA and Open CL interfaces are being progressively developed. There are over 500 algorithms and about 10 times functions that form or back those algorithms. Open CV is written inherently in C++ and has a template interface that works harmoniously with STL containers.

**IDLE**

IDLE is Pythons Integrated Development and Learning Environment. IDLE is completely coded in Python, using the tkinter GUI toolkit. It works mostly uniformly on Windows, Unix and macOS. It has a Python shell window (interactive interpreter) with colorizing of error messages, code input and code output. There is a multi-window text editor with multiple undo, Python colorizing, smart indent, call tips, auto completion, and other features. Searching within any window, replacing within editor windows and searching through multiple files is possible. It also has configuration, browsers and other dialogs as well.

**CHAPTER-5**

**CODING**

import math

import numpy as np

import cv2

from matplotlib import pyplot as plt

from PIL import Image, ImageFilter

from PIL import Image

from PIL import Image

%matplotlib inline

im1=cv2.imread('E:/datasets/20/252.png')

cv2.imshow("Input Image", im1) // Show the input image

im2= cv2.cvtColor(im1, cv2.COLOR\_BGR2GRAY) // Convert RGB to GRAY

cv2.imshow("GrayScale" ,im2)

cv2.waitKey()

**Resize the Image**

resized = cv2.resize(im2, (2000, 1500))

print(f"Resized Dimensions : {resized.shape}")

cv2.imwrite('resized\_imaged.jpg', resized)

cv2.imshow("Input Image", resized)

cv2.waitKey()

**Speckle Noise**

image = cv2.cvtColor(im1 , cv2.COLOR\_BGR2GRAY)

dst = cv2.fastNlMeansDenoisingColored(im1,None,10,10,7,21)

plt.subplot(121),plt.imshow(im2)

plt.subplot(122),plt.imshow(dst)

plt.show()

hist,bins = np.histogram(dst.flatten(),256,[0,256])

cdf = hist.cumsum()

cdf\_normalized = cdf \* hist.max()/ cdf.max()

plt.plot(cdf\_normalized, color = 'b')

plt.hist(dst.flatten(),256,[0,256], color = 'r')

plt.xlim([0,256])

plt.legend(('cdf','histogram'), loc = 'upper left')

plt.show()

def calculate\_psnr(im1,dst):

im1 = im1.astype(np.float64)

dst= dst.astype(np.float64)

mse = np.mean((im1 - dst)\*\*2)

if mse == 0:

return float('inf')

return 20 \* math.log10(255.0 / math.sqrt(mse))

d=calculate\_psnr(im1,dst)

print("The PSNR value of Speckle noise is ",d)

**Poisson Noise**

image = cv2.cvtColor(im1 , cv2.COLOR\_BGR2GRAY)

gauss = np.random.normal(0,1,im1.size)

gauss = gauss.reshape(im1.shape[0],im1.shape[1],im1.shape[2]).astype('uint8')

noise = im1 + im1 \* gauss

cv2.imshow('a',noise)

cv2.waitKey(0)

hist,bins = np.histogram(noise.flatten(),256,[0,256])

cdf = hist.cumsum()

cdf\_normalized = cdf \* hist.max()/ cdf.max()

plt.plot(cdf\_normalized, color = 'b')

plt.hist(noise.flatten(),256,[0,256], color = 'r')

plt.xlim([0,256])

plt.legend(('cdf','histogram'), loc = 'upper left')

plt.show()

def calculate\_psnr(im1, noise):

im1 = im1.astype(np.float64)

noise= noise.astype(np.float64)

mse = np.mean((im1 - noise)\*\*2)

if mse == 0:

return float('inf')

return 20 \* math.log10(255.0 / math.sqrt(mse))

d=calculate\_psnr(im1,noise)

print("The PSNR value of poisson Noise is:",d)

**Gaussian Noise**

image = cv2.cvtColor(im1 , cv2.COLOR\_BGR2GRAY)

gauss = np.random.normal(0,1,im1.size)

gauss = gauss.reshape(im1.shape[0],im1.shape[1],im1.shape[2]).astype('uint8')

img\_gauss = cv2.add(im1,gauss)

cv2.imshow('a',img\_gauss)

cv2.waitKey(0)

hist,bins = np.histogram(img\_gauss.flatten(),256,[0,256])

cdf = hist.cumsum()

cdf\_normalized = cdf \* hist.max()/ cdf.max()

plt.plot(cdf\_normalized, color = 'b')

plt.hist(img\_gauss.flatten(),256,[0,256], color = 'r')

plt.xlim([0,256])

plt.legend(('cdf','histogram'), loc = 'upper left')

plt.show()

def calculate\_psnr(im1, img\_gauss):

im1 = im1.astype(np.float64)

img\_gauss= img\_gauss.astype(np.float64)

mse = np.mean((im1 - gauss)\*\*2)

if mse == 0:

return float('inf')

return 20 \* math.log10(255.0 / math.sqrt(mse))

d=calculate\_psnr(im1,img\_gauss)

print(" PSNR value of Gaussian Noise is:",d)

**Mean Filter**

image = cv2.cvtColor(im1 , cv2.COLOR\_BGR2HSV)

figure\_size = 9 # the dimension of the x and y axis of the kernal.

new\_image = cv2.blur(image,(figure\_size, figure\_size))

plt.figure(figsize=(11,6))

plt.subplot(121),plt.imshow(cv2.cvtColor(image,cv2.COLOR\_HSV2RGB)),

plt.title('Original')

plt.xticks([]), plt.yticks([])

plt.subplot(122),plt.imshow(cv2.cvtColor(new\_image,cv2.COLOR\_HSV2RGB)),

plt.title('Mean filter')

plt.xticks([]), plt.yticks([])

plt.show()

hist,bins = np.histogram(new\_image.flatten(),256,[0,256])

cdf = hist.cumsum()

cdf\_normalized = cdf \* hist.max()/ cdf.max()

plt.plot(cdf\_normalized, color = 'b')

plt.hist(new\_image.flatten(),256,[0,256], color = 'r')

plt.xlim([0,256])

plt.legend(('cdf','histogram'), loc = 'upper left')

plt.show()

def calculate\_psnr(image, new\_image):

image = image.astype(np.float64)

new\_image= new\_image.astype(np.float64)

mse = np.mean((image - new\_image)\*\*2)

if mse == 0:

return float('inf')

return 20 \* math.log10(255.0 / math.sqrt(mse))

d=calculate\_psnr(image,new\_image)

print(" PSNR value of Mean Filter is:",d)

**Gaussain Filter**

new\_image = cv2.GaussianBlur(im1, (figure\_size, figure\_size),0)

plt.figure(figsize=(11,6))

plt.subplot(121),plt.imshow(cv2.cvtColor(image,cv2.COLOR\_HSV2RGB)),

plt.title('Original')

plt.xticks([]), plt.yticks([])

plt.subplot(122),plt.imshow(cv2.cvtColor(new\_image,cv2.COLOR\_HSV2RGB)),

plt.title('Gaussian Filter')

plt.xticks([]), plt.yticks([])

plt.show()

hist,bins = np.histogram(new\_image.flatten(),256,[0,256])

cdf = hist.cumsum()

cdf\_normalized = cdf \* hist.max()/ cdf.max()

plt.plot(cdf\_normalized, color = 'b')

plt.hist(new\_image.flatten(),256,[0,256], color = 'r')

plt.xlim([0,256])

plt.legend(('cdf','histogram'), loc = 'upper left')

plt.show()

def calculate\_psnr(image,new\_image ):

image = image.astype(np.float64)

new\_image= new\_image.astype(np.float64)

mse = np.mean((image - new\_image)\*\*2)

if mse == 0:

return float('inf')

return 20 \* math.log10(255.0 / math.sqrt(mse))

d=calculate\_psnr(image,new\_image)

print(" PSNR value of Gaussian Filter is:",d)

**Median Filter**

new\_image = cv2.medianBlur(im1, figure\_size)

plt.figure(figsize=(11,6))

plt.subplot(121),plt.imshow(cv2.cvtColor(image,cv2.COLOR\_HSV2RGB)),

plt.title('Original')

plt.xticks([]), plt.yticks([])

plt.subplot(122),plt.imshow(cv2.cvtColor(new\_image,cv2.COLOR\_HSV2RGB)),

plt.title('Median Filter')

plt.xticks([]), plt.yticks([])

plt.show()

hist,bins = np.histogram(new\_image.flatten(),256,[0,256])

cdf = hist.cumsum()

cdf\_normalized = cdf \* hist.max()/ cdf.max()

plt.plot(cdf\_normalized, color = 'b')

plt.hist(new\_image.flatten(),256,[0,256], color = 'r')

plt.xlim([0,256])

plt.legend(('cdf','histogram'), loc = 'upper left')

plt.show()

def calculate\_psnr(image, new\_image):

image = image.astype(np.float64)

new\_image= new\_image.astype(np.float64)

mse = np.mean((image- new\_image)\*\*2)

if mse == 0:

return float('inf')

return 20 \* math.log10(255.0 / math.sqrt(mse))

d=calculate\_psnr(image,new\_image)

print(" PSNR value of Median Filter is:",d)

**Canny-Edge**

t\_lower = 50

t\_upper = 150

edge = cv2.Canny(im1, t\_lower, t\_upper)

hist,bins = np.histogram(edge.flatten(),256,[0,256])

cdf = hist.cumsum()

cdf\_normalized = cdf \* hist.max()/ cdf.max()

plt.plot(cdf\_normalized, color = 'b')

plt.hist(edge.flatten(),256,[0,256], color = 'r')

plt.xlim([0,256])

plt.legend(('cdf','histogram'), loc = 'upper left')

plt.show()

def calculate\_psnr(im1,edge):

im1= im1.astype(np.float64)

edge= edge.astype(np.float64)

mse = np.mean((im1-edge)\*\*2)

if mse == 0:

return float('inf')

return 20 \* math.log10(255.0 / math.sqrt(mse))

d=calculate\_psnr(im1,gauss)

print(" PSNR value of canny edge is:",d)

cv2.imshow('Original', im1)

cv2.imshow('edge', edge)

cv2.waitKey(0)

cv2.destroyAllWindows()

**Laplacian**

laplacian = cv2.Laplacian(im1, cv2.CV\_64F)

plt.subplot(2,2,1),plt.imshow(im1,cmap = 'gray')

plt.title('Original'), plt.xticks([]), plt.yticks([])

plt.subplot(2,2,2),plt.imshow(laplacian,cmap = 'gray')

plt.title('Laplacian'), plt.xticks([]), plt.yticks([])

plt.show()

hist,bins = np.histogram(laplacian.flatten(),256,[0,256])

cdf = hist.cumsum()

cdf\_normalized = cdf \* hist.max()/ cdf.max()

plt.plot(cdf\_normalized, color = 'b')

plt.hist(laplacian.flatten(),256,[0,256], color = 'r')

plt.xlim([0,256])

plt.legend(('cdf','histogram'), loc = 'upper left')

plt.show()

def calculate\_psnr(im1, laplacian):

im1 = im1.astype(np.float64)

laplacian= laplacian.astype(np.float64)

mse = np.mean((im1 - laplacian)\*\*2)

if mse == 0:

return float('inf')

return 20 \* math.log10(255.0 / math.sqrt(mse))

d=calculate\_psnr(im1,laplacian)

print(" PSNR value of laplacian is:",d)

**Gradient**

sobelx = cv2.Sobel(im1,cv2.CV\_64F,1,0,ksize=5)

sobely = cv2.Sobel(im1,cv2.CV\_64F,0,1,ksize=5)

plt.subplot(2,2,1),plt.imshow(im1,cmap = 'gray')

plt.title('Original'), plt.xticks([]), plt.yticks([])

plt.subplot(2,2,3),plt.imshow(sobelx,cmap = 'gray')

plt.title('Sobel X'), plt.xticks([]), plt.yticks([])

plt.subplot(2,2,4),plt.imshow(sobely,cmap = 'gray')

plt.title('Sobel Y'), plt.xticks([]), plt.yticks([])

plt.show()

hist,bins = np.histogram(sobelx.flatten(),256,[0,256])

cdf = hist.cumsum()

cdf\_normalized = cdf \* hist.max()/ cdf.max()

plt.plot(cdf\_normalized, color = 'b')

plt.hist(sobelx.flatten(),256,[0,256], color = 'r')

plt.xlim([0,256])

plt.legend(('cdf','histogram'), loc = 'upper left')

plt.show()

def calculate\_psnr(im1, sobelx):

im1 = im1.astype(np.float64)

sobelx= sobelx.astype(np.float64)

mse = np.mean((im1 - sobelx)\*\*2)

if mse == 0:

return float('inf')

return 20 \* math.log10(255.0 / math.sqrt(mse))

d=calculate\_psnr(im1,sobelx)

print(" PSNR value of sobelx is:",d)

hist,bins = np.histogram(sobely.flatten(),256,[0,256])

cdf = hist.cumsum()

cdf\_normalized = cdf \* hist.max()/ cdf.max()

plt.plot(cdf\_normalized, color = 'b')

plt.hist(sobely.flatten(),256,[0,256], color = 'r')

plt.xlim([0,256])

plt.legend(('cdf','histogram'), loc = 'upper left')

plt.show()

def calculate\_psnr(im1, sobely):

im1 = im1.astype(np.float64)

sobely= sobely.astype(np.float64)

mse = np.mean((im1 - sobely)\*\*2)

if mse == 0:

return float('inf')

return 20 \* math.log10(255.0 / math.sqrt(mse))

d=calculate\_psnr(im1,sobely)

print(" PSNR value of sobely is:",d)

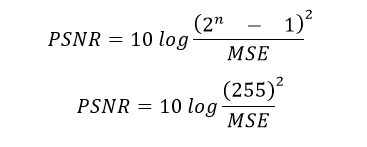
https://homepages.inf.ed.ac.uk/rbf/HIPR2/mote.gif

https://homepages.inf.ed.ac.uk/rbf/HIPR2/mote.gif

**CHAPTER-6**

**PERFORMANCE METRICS**

The peak signal-to-noise ratio (PSNR) is the ratio between a signal's maximum power and the power of the signal's noise.  PSNR is utilized to measure the nature of recreation of lossy and lossless compression and addition of different kind unwanted noise values. When looking at compression codes, PSNR is an estimate of human impression of reconstruction quality. A higher PSNR for the most part shows that the reformation is of higher standard, now and again it may not. For concealing pictures with three RGB esteems for every pixel, the importance of PSNR is the comparable beside the MSE is the absolute over totally squared worth differentiations isolated by picture size and by three. Here we have mainly focused on three filtering techniques (Mean filter, Gaussian filter, Median filter) of three noise models i.e., gaussian, speckle and poisson and three segmentation are canny-edge, laplacian and gradient. The PSNR described below:



For image quality measurement, if the value of PSNR is very high for an image of a particular noise type then it is defined as a best quality image.

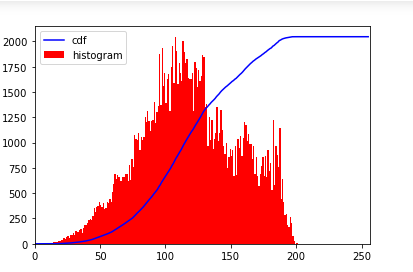
To estimate the PSNR of an image, it is necessary to compare that image to an ideal clean image with the maximum possible power.

**Performance Analysis on Different Types of Noises**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **TYPES OF NOISES** | **PSNR VALUES** |
| **1.** | **Speckle Noise** | **28.31** |
| **2.** | **Poisson Noise** | **12.42** |
| **3.** | **Gaussian Noise** | **6.025** |

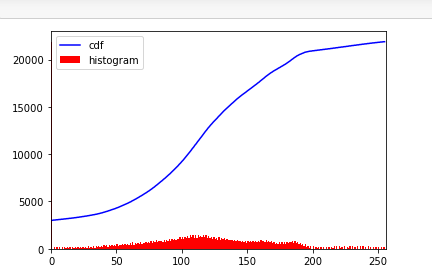
**SPECKLE NOISE**

The histogram is represented by the speckle noise.



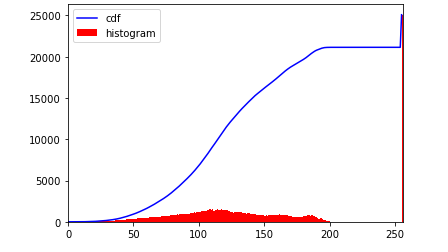
**POISSON NOISE**

The histogram is represented by the poisson noise.

****

**GAUSSIAN NOISE**

The histogram is represented by the Gaussian noise.

. 

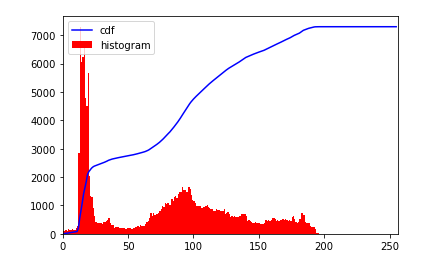
In currency image of the speckle noise aggregate value is 28.31 and the poisson noise aggregate value is 12.48 and Gaussian noise aggregate value is 6.025 . Here 28.31 > 12.48 > 6.025. so speckle noise has given better PSNR values result than poisson and Gaussian noise based on the table.

**Performance Analysis on Different Types of Filters**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **TYPES OF FILTERS** | **PSNR VALUES** |
| **1.** | **Mean Filter** | **24.37** |
| **2.** | **Gaussian Filter** | **12.02** |
| **3.** | **Median Filter** | **11.78** |

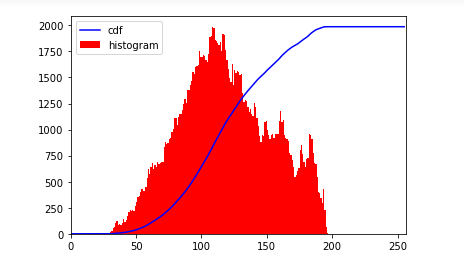
**MEAN FILTER**

The histogram is represented by the Mean Filter.



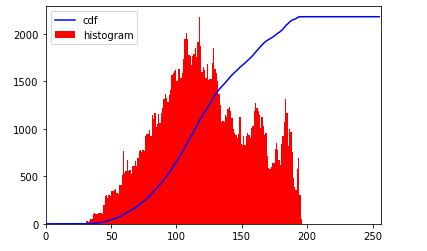
**GAUSSIAN FILTER**

The histogram is represented by the Gaussian Filter.



**MEDIAN FILTER**

The histogram is represented by the Median Filter.

****

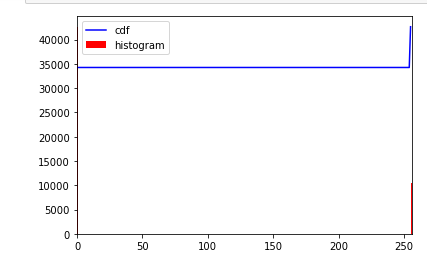
In currency image of the Mean Filter aggregate value is 24.37 and the Gaussian Filter aggregate value is 12.02 and Median Filter aggregate value is 11.78. Here 24.37 > 12.02 > 11.78. So Mean Filter has given better PSNR values result than Gaussian and Median Filter based on the table.

**Performance Analysis on Segmentation**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **SEGMENTATION** | **PSNR VALUES** |
| **1.** | **Canny-Edge** | **6.025** |
| **2.** | **Laplacian** | **5.461** |
| **3.** | **Gradient**  **Sobel x**  **Sobel y** | **-9.076**  **-9.530** |

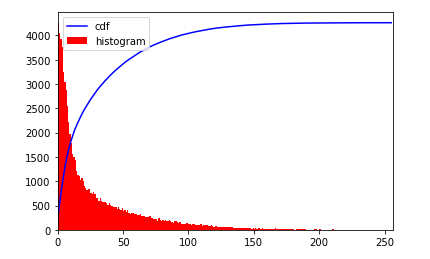
**CANNY-EDGE**

The histogram is represented by the Canny-Edge is.



**LAPLACIAN**

The histogram is represented by the Laplacian is.



### GRADIENT

### The histogram is represented by the Gradient(Sobel x,Sobel y) is.

### SOBEL X

### 

### SOBEL Y

### 

In currency image of the Canny-Edge aggregate value is 6.025 and the Laplacian aggregate value is 5.461 and Sobel x aggregate value is -9.076 and Sobel y aggregate value is -9.530. Here 6.025 > 5.461 > -9.076 > -9.530. So Canny-Edge has given better PSNR values result than Laplacian and Gradient based on the table.

### 

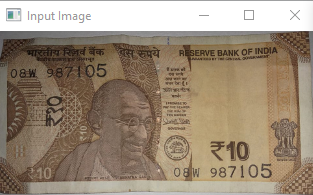
### CHAPTER – 7

### RESULT AND DISCUSSION

All the approaches described in the previous chapter, namely, Noises, Filters, Canny-Edge, Laplacian and Gradient approach are applied on several images. The results are shown in below. Generally speaking, conclusions regarding the performance of these methods can be made by observing the results.

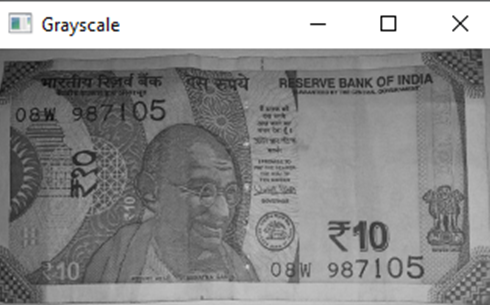
**1. INPUT IMAGE**

Below figure shows the input image.



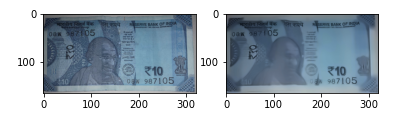
**2. GRAY SCALE CONVERSION**

It is easy to remove the noise by using conversion of colours images to gray scale images. Below figure shows the Gray scale image.

\

**3. SPECKLE NOISE**

Speckle noise is a one type of noise technique. Below figure shows the speckle noise.

****

**4. POISSON NOISE**

Poisson noise is a one type of noise technique. Below figure shows the poisson noise.

****

### 5.GAUSSIAN NOISE

Gaussian noise is a one type of noise technique. Below figure shows the Gaussian noise.



**6. MEAN FILTER**

Mean Filter is a one type of noise removing technique. Below figure shows the Mean Filter.



**7. GAUSSIAN FILTER**

Gaussian Filter is a one type of noise removing technique. Below figure shows the Gaussian Filter.



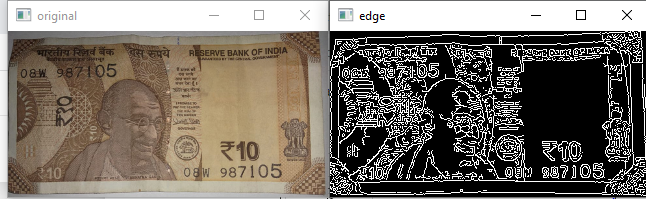
**8.MEDIAN FILTER**

Median Filter is a one type of noise removing technique. Below figure shows the Median Filter.

****

### 9. CANNY-EDGE

Canny-Edge is a one type of segmentation process. Below figure shows the Canny-Edge.



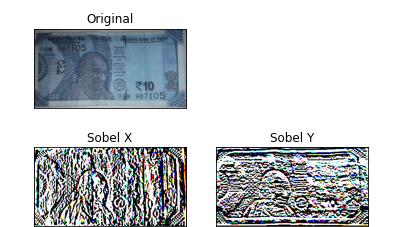
**10. LAPLACIAN**

Laplacian is a one type of segmentation process. Below figure shows the Laplacian.



**11. GRADIENT**

Gradient is a one type of segmentation process. Below figure shows the Gradient



### 

### 

### 

### 

### CHAPTER-8

### CONCLUSION AND FUTURE SCOPE

### 8.1 CONCLUSION

Currency use is a necessity for survival and hence it is always necessary to keep in track of its originality. Paper currencies are used much more in India and hence a system to detect the fake currency is needed. A technique for verifying Indian paper currency. The approach gives an efficient method of fake currency detection based on physical appearance. The work will surely be very useful for minimizing the counterfeit currency. Through this application are able to see the missing parameters which the fake note doesn’t have as compared to the original notes. Original Currency is being detected using Image Processing Technique. Noise removal is one the most significant pre-processing method to improve the quality of picture from noised input picture. Here, two separating procedure are implemented to currency images are various kind of Noise for currency pictures. The performance metrics determines the efficiency of the algorithm used. This paper investigated and analyzed three filtering methods and segmentation process. Investigational opinion under PSNR shows the efficient filter and segmentation performance at various noises. A relative study has been motivated among those filters to successfully de-noising the images and segment the images. The conclusion of this study exposes that speckle noise provides relatively better performance for reduction among the other noises in terms of PSNR values. Besides Mean filter noise removal performances comparatively better than the median and Gaussian filters. And Canny result performances comparatively better than the laplacian and gradient. Canny result is superior one when compared to all for a selected image since different edge detections work better under different conditions.

**8.2 FUTURE SCOPE**

In future this work will be extended as to apply the modified deep CNN (Convolutional Neural Network) classification to compare the original or forgery currency. The model's accuracy will be improved by experimenting with modified deep CNN architectures in the future. Increasing the data set for the model to be better trained and give better outcomes. Applying image pre-processing techniques like noise removal and edge-detection to crop the currency note out of an image will present a better input to the deep CNN. To build on this, we propose to form a much larger data set with real-world like pictures of real and fake currency notes. This will help in providing a much more realistic model. Moreover, with a large data set available, deep learning algorithms like Convolutional Neural Networks or CNN can be applied which have high accuracy in image processing scenarios. Furthermore, by using CNN the project can directly analyze images as input, and wavelet transformation will not be required. This can make the system more convenient and user friendly to use. Moreover, as it is likely to be used in financial institutions it will be more convenient for users to directly click a picture and get it verified, this can be done with the help of CNN as mentioned above. Hence, to make the project more robust and professional the above suggest measures can be implemented.

**REFERENCES**

[1] M.N.Rathore and J.Sagar, “A Review on Fake currency detection using feature extraction,” vol. 10, no. 11, pp. 407–411, 2019.

[2] S.Adhikari, S.Thapa, and B.K.Shah, “Oversampling based Classifiers for Categorization of Radar Returns from the Ionosphere,” Proc. Int. Conf. Electron. Sustain. Commun. Syst. ICESC 2020, no. Icesc, pp. 975–978, 2020, doi: 10.1109/ICESC48915.2020.9155833.

[3] A.Ghimire, S.Thapa, A.K.Jha, S.Adhikari, and A.Kumar, “Accelerating business growth with big data and artificial intelligence,” Proc. 4th Int. Conf. IoT Soc. Mobile, Anal. Cloud, ISMAC 2020, pp. 441–448, 2020, doi: 10.1109/ISMAC49090.2020.9243318.

[4] Reserve Bank of India, http://rbi.org.in/scripts/icbanknotesecurity.aspx

[5] The Financial Express, https://www.financialexpress.com/economy/1139640

[6] S. V. Viraktamath; Kshama Tallur; Rohan Bhadavankar; Vidya, “Review on Detection of Fake Currency using Image processing Techniques”, 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS).

[7] Kiran Kamble; Anuthi Bhansali; Pranali Satalgaonkar; Shruti Alagundgi, “Counterfeit Currency Detection using Deep Convolutional Neural Network”, 2019 IEEE Pune Section International Conference (PuneCon).

[8] Gajanand Gupta, “Algorithm for Image Processing Using Improved Median Filter and Comparison of Mean and Improved Median Filter” (IJSCE) ISSN: 2231-2307, Volume-1, Issue-5, November 2021.

[9] NeeruRathee, ArunKadian, RajatSachdeva “Feature Fusion for Fake Indian Currency Detection”.