2. IMPULSE NOISE MODELS

In this chapter, both SPIN and RVIN impulse noise models are proposed. Correctness of the proposed noise models are validated by implementing the algorithms and analyzing the output graphs and observing the changes in the Mean Absolute Error (MAE) values. This chapter is organized as follows. Section 2.1 discusses different types of impulse noise; the problem definition is provided in Section 2.2, scope of the research work is discussed in Section 2.3, performance measurements are presented in Section 2.4 followed by simulations results in Section 2.5. Section 2.6 provides the chapter summary.

2.1 Impulse Noise Types

![Classification of Image Noise Diagram]

Figure 2.1: Classification of Image Noise

Noise is any unwanted signal present in the original signal. Noise is broadly classified into two main categories: blur noise and impulse noise, as shown in Figure 2.1. Blur noise is a uniform noise which blurs the image. It modifies all the pixels present in the image by shifting pixel values towards low or high intensity levels of the image (Gaussian Blur). Sometimes, the noise blurs the image in a particular direction which is then known as motion blur. Main causes of blur noise are atmospheric turbulence, missed focus of camera lens, improper opening and closing of camera shutter and relative motion between cameras and the object. Impulse noise is a non-uniformly distributed noise. It modifies only select pixels from the image keeping the remaining pixels unchanged. Impulse noise produces dot spots or patches on the image.

\[
X'_{ij} = \begin{cases} 
N_{ij} & \text{with probability } p \\
X_{ij} & \text{With probability } (1-p) 
\end{cases}
\]  

(2.1)
Impulse noise has the property of either leaving pixels unmodified with probability \((1 - p)\) or replacing it all together with a probability of \(p\), as shown in Equation (2.1). Outputs of Gaussian blur and impulse noises are shown in Figure 2.2.

![Figure 2.2: Results of Gaussian and Impulse Noise](image)

The sources of impulse noise are usually the result of error in transmission system, faulty sensors present in storage or capturing devices, and atmospheric or man-made disturbances. Impulse noise is the most import issue in television and satellite transmission systems.

![Figure 2.3: Types of Impulse Noise](image)

Based on the type of value it adds to image, impulse noise is classified into two types: SPIN and RVIN, as shown in Figure 2.3. SPIN adds fixed value to all corrupted pixels,
possibly low or high values. SPIN adds possibly a low value 0 and a high value 255. RVIN adds random values and the possible values of pixel intensity vary between 0 and 255.

### 2.1.1 Salt and Pepper Impulse Noise Model

![Input Image](image1)

![Corrupted Image with 30% SPIN](image2)

**Figure 2.4: Addition of SPIN using algorithm 2.1.**

\[
Y_{ij} = \begin{cases} 
X_{ij} & \text{non-corrupted pixel with probability } (1 - p) \\
[0 \text{ or } 255] & \text{corrupted pixel with Probability } p 
\end{cases} 
\]  

(2.2)

where

- \( Y_{ij} \rightarrow (i, j)^{th} \) pixel value of corrupted image. \( X_{ij} \rightarrow (i, j)^{th} \) pixel value of original image.
- \( P \rightarrow \) Impulse noise ratio.
- \( i \) and \( j \) → Coordinate values of pixel.

SPIN is also known as Fixed Valued Impulse Noise because it possesses fixed noise values either 0 or 255. In SPIN, a noisy pixel can have only two possible values: minimum intensity level of 0 in gray scale images for black color (pepper noise) or maximum inten-
sity level of 255 in gray scale images for white color (salt noise), as mentioned in Equation (2.2). To add SPIN to image, the values 0 or 255 are used as replacement values for the corrupted pixels, as shown in Figure 2.4. To introduce required percentage of SPIN into an image, we randomly select required percentage of pixels from given input image and replace their values randomly by values 0's and 255's, as shown in Algorithm 2.1. Table 2.1 and Figure 2.7 show the implemented outputs of SPIN algorithm.

Algorithm 2.1- To introduce SPIN into an image

---

**Input:**

$I[1:x,1:y] \rightarrow$ Input gray scale image contains integer values from 0 to 255.

$N_r \rightarrow$ Noise percentage.

**Output:**

$In[1:x,1:y] \rightarrow$ Noise added gray scale image contains integer values from 0 to 255.

**Initialization:**

$N[1:x,1:y] \rightarrow$ Matrix contains 0's or 255's values randomly in randomly selected locations.

$N_1[1:x,1:y] \rightarrow$ Binary Matrix contains $N_r$ percentage of 1's in randomly selected locations.

**Procedure:**

1. Start
2. $[x, y] \leftarrow$ Size(I)
3. For i ← 1:x:1
4.   For j ← 1:y:1
5.     If $(N_1(i,j) == 1) \rightarrow$ $In(i,j) \leftarrow N(i,j)$
6.     Else
7.       $In(i,j) \leftarrow I(i,j)$
8.     End
9.   End
10. End
11. End
12. Return (In)
13. Stop

---

**Table 2.1: MAE (dB) for Different Percentage of SPIN**

<table>
<thead>
<tr>
<th>Noise</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAE</td>
<td>12.38</td>
<td>24.63</td>
<td>37.22</td>
<td>49.10</td>
<td>61.82</td>
<td>73.84</td>
<td>86.49</td>
<td>98.96</td>
<td>111.06</td>
</tr>
</tbody>
</table>
Table 2.1 shows the effect of adding SPIN using Algorithm 2.1 for different noise percentages. From the table, we can observe that, as the noise percentage increases, the Mean Absolute Error (MAE) value increases. MAE is an error measurement which is the difference between original input image and the noise added image, as shown in Equation (2.4). Hence, for higher values of MAE the difference between original image and noise added image increases. Increase in MAE for an increase in noise percentage shows that the proposed SPIN noise adding algorithm works correctly and that the noise gets added correctly to the image.

2.1.2 Random Valued Impulse Noise Model

![](image)

**Figure 2.5: Addition of RVIN using algorithm 2.2.**

RVIN is known as Random Valued Impulse Noise because it possesses any noise values between 0 to 255, as mentioned in Equation (2.3). A noisy pixel can have all possible intensity values from minimum to maximum [0 to 255]. To add RVIN to the image, the val-
ues 0 to 255 are used as noise values for the corrupted pixel, as shown in Figure 2.5. To introduce required percentage of RVIN into an image, we randomly select required percentage of pixels from given input image and replace their values from randomly selected values between 0 and 255, as shown in Algorithm 2.2. Table 2.2 and Figure 2.8 show the implemented outputs of RVIN algorithm.

\[ Y_{ij} = \begin{cases} X_{ij} & \text{non-corrupted pixel with probability } (1-p) \\ [0 \text{ to } 255] & \text{corrupted pixel with Probability } p \end{cases} \quad (2.3) \]

where

\( Y_{ij} \rightarrow (i, j)^{th} \) pixel value of corrupted image. \( X_{ij} \rightarrow (i, j)^{th} \) pixel value of original image.

\( P \rightarrow \) Impulse noise ratio. \( i \) and \( j \rightarrow \) Coordinate values of pixel.

```
Algorithm 2.2- To introduce RVIN into an image
```

**Input:**

\( I[1:x,1:y] \rightarrow \) Input gray scale image contains integer values from 0 to 255.

\( Nr \rightarrow \) Noise percentage.

**Output:**

\( In[1:x,1:y] \rightarrow \) Noise added gray scale image contains integer values from 0 to 255.

**Initialization:**

\( N[1:x,1:y] \rightarrow \) Matrix contains 0 to 255 values randomly in randomly selected locations.

\( N1[1:x,1:y] \rightarrow \) Binary Matrix contains \( Nr \) percentage of 1's in randomly selected locations.

**Procedure:**

1. Start
2. \([x, y] \leftarrow \text{Size}(I)\)
3. For \( i \leftarrow 1:x:1 \)
4. \hspace{1cm} For \( j \leftarrow 1:y:1 \)
5. \hspace{2cm} If \( (N1(i,j)=1) \)
6. \hspace{3cm} \( \text{In}(i,j) \leftarrow N(i,j) \)
7. \hspace{2cm} Else
8. \hspace{3cm} \( \text{In}(i,j) \leftarrow I(i,j) \)
9. \hspace{2cm} End
10. \hspace{1cm} End
11. End
12. Return \( (In) \)
13. Stop

---
Table 2.2: MAE (dB) for Different Percentage of RVIN

<table>
<thead>
<tr>
<th>Noise</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAE</td>
<td>08.81</td>
<td>17.69</td>
<td>26.48</td>
<td>36.05</td>
<td>44.53</td>
<td>53.41</td>
<td>62.06</td>
<td>71.34</td>
<td>80.01</td>
</tr>
</tbody>
</table>

Table 2.2 shows the effect of adding RVIN using Algorithm 2.2 for different noise percentages. From the table, we can observe that, as the noise percentage increases, the Mean Absolute Error (MAE) value increases. MAE is an error measurement which is the difference between original input image and noise added image, as mentioned in Equation (2.4). Hence, for higher values of MAE the difference between original image and noise added image increases. Increase in MAE for an increase in noise percentage shows that the proposed RVIN noise adding algorithm works correctly and that the noise gets added correctly to the image.

2.2 Problem Definition

Digital image processing algorithms play a vital role in all the fields of science, engineering and technology. Performance of algorithms basically depends on the quality of the input image supplied to the algorithms. If the quality of input image is good then the quality of output is also good else we get low quality output. Efficiency of all image processing algorithms is directly proportional to the quality of the input image. Hence, image quality enhancement or image de-noising techniques for images corrupted by various types of noise are one of the most important issues in digital image processing.

Image noises can be broadly classified into two types: Gaussian blur noise and impulse noise. Gaussian blur noise is a uniform noise which affects all pixels uniformly where as impulse noise is non-uniform noise which affects only select pixels in an image. Image blurring is one of the prime causes of poor image quality in digital imaging. Two main causes of blurry images are out-of-focus and camera shake. Impulse noise is one which may corrupt the images during their acquisition or transmission or storage etc. Several algorithms are proposed in the literature to remove impulse noise in the images. Some algorithms provide good results in low noise conditions and weak results in high noise conditions and vice versa. Further, such algorithms are not well-suited for real world applications to remove noise since they require prior knowledge of percentage of noise present in the image. This information is not available in real world scenarios. In our research, both fixed (salt and pepper) and variable (random valued) impulse noise models
are considered for image de-noising. The SPIN assumes a minimum value of 0 and a maximum value of 255 of noise and RVIN method assumes a noise value between minimum of 0 and maximum of 255. The primary goal of this research is to design efficient high speed de-noising algorithms for images corrupted by different types of impulse noise, which produces consistent outputs in both low and high noise conditions without any assumption about the image noise level in an algorithm. Such algorithms can be used in real-time applications without any modifications for different noise percentages. Further, these algorithms have to produce consistent outputs in both low and high noise conditions within a small processing time so that the algorithms meet the real-time constraints.

2.3 Scope of Research

Our research focuses on both types of impulse noise: Salt and Pepper Impulse Noise and Random Valued Impulse Noise. We develop impulse noise reduction techniques for gray scale images. Gray scale algorithms can very easily be extended to color image processing and video processing because color images can be considered as a set of three gray scale images Red, Green and Blue. Similarly video can be considered as an image sequence. Algorithms developed for single images can be repeated for video sequences as well.

2.4 Performance Measurements

\[
MAE(C) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} |(X_{ij} - C_{ij})|}{M \times N}
\]

\[
ERP = \left(1 - \frac{MAE(\text{restored image})}{MAE(\text{corrupted image})}\right) \times 100
\]

where

X → Original image.  \hspace{1cm} ERP → Error Recovery Percentage.
M×N → Size of the image.  \hspace{1cm} MAE(C) → Mean Absolute Error of image C.
i → Integer variable changes from 1 to M.  \hspace{1cm} j → Integer variable changes from 1 to N.
C → Image which is compared with original image.
Figure 2.6: MAE and ERP calculations.

To evaluate the performance of impulse noise algorithms the performance measure Error Recovery Percentage (ERP) is used. ERP is measured using unit decibel (dB). ERP is the amount of Mean Absolute Error (MAE) recovered by an algorithm. The MAE gives the difference between the given two input images, as shown in Equation (2.4). Low value of MAE indicates more similarity between the given images and vice versa. MAE value ranges from 0 to 255. Zero value of MAE indicates both images look exactly the same.
and as MAE value increases towards 255 the similarity between images decreases. ERP calculated by using percentage of the difference of corrupted image MAE and restored image MAE with respect to corrupted image MAE as shown in Equation (2.5). The ERP is the percentage of noise recovered by an algorithm. Maximum value of ERP is 100 which indicates that both the original and the restored images are exactly the same; this also means that the restoration is 100%; i.e. the algorithm has successfully recovered all corrupted pixels. Sometimes, the algorithms return negative value of ERP indicating that the algorithm increases the noise instead of restoring the image. For good de-noising algorithms the restored image MAE is less than the corrupted image MAE; a negative ERP value indicates the de-noising algorithm is bad. A common measuring unit ERP is used for both SPIN and RVIN. With the increase in noise [44], the MAE value increases. Since ERP is especially designed to measure and to compare the performance of image restoration filters, MAE value of corrupted image should not be 0. Since, in all filters, recovering corrupted pixels is done using uncorrupted neighboring pixels, it is not possible to recover the image if it is 100% corrupt. ERP can be used to check restoration results from 1% to 100% noise. For 0% noise, MAE can be used to measure and to compare the performance of image restoration filters. Calculations of MAE and ERP are shown in Figure 2.6.

2.5 Simulations and Results

Simulations were carried out on a system with 4GB RAM, Intel ® Core(TM) i5-4210U CPU @ 1.70GHz 2.40 GHz processor, Windows 7 64-bit operating system and MatLab-R2009a version. Quality of SPIN model is measured by observing changes in the MAE value with respect to the noise. The MAE value increases with increase in noise and vice versa and the MAE is calculated using Equation (2.4). Figure 2.9 shows changes in MAE with changes in SPIN percentage and we observe that the MAE value increases with increase in SPIN level. Hence, by observing the decrease in the visibility quality and the increase in MAE values with SPIN percentage of Figure 2.7 and Figure 2.9, respectively, we conclude that the implemented SPIN model works correctly and successfully by adding the required SPIN percentage. Similar to SPIN model, by observing the decrease in the visibility quality and the increase in the MAE values with SPIN percentage of Figure 2.8 and Figure 2.10, respectively, we conclude that RVIN model effectively adds required noise percentage to the input image.
2.6 Summary

Error in transmission system, faulty sensors present in the storage or capturing devices and atmospheric or man-made disturbances are the most commonly occurring impulse noise in digital image processing. Impulse noise is classified into Salt and Pepper Impulse Noise (SPIN) and Random Valued Impulse Noise (RVIN). In this chapter, both types of impulse noise models SPIN and RVIN are proposed. Both models are implemented and analyzed by giving different values of noise percentage and our experimental results show that both the models work correctly.
Figure 2.7: Results for Different Percentage of SPIN
Figure 2.8: Results for Different Percentage of RVIN
Figure 2.9: MAE vs. SPIN Graph

Figure 2.10: MAE vs. RVIN Graph