Neutrosophic Sets and Systems, Vol. 67, 2024
University of New Mexico

TripleMAsk Spatial Linear Filter and Neutrosophic Entropy for Video Denoising, Face Detection and Recognition in Forensic Crime Analysis Using Deep Learning

Anita Sigamani [1], Dr. Prema Selvaraj [2]

[1] Department of Computer Science, Bhusanayana Mukundadas Sreenivasaiah College for Women, Bengaluru, Karnataka, India; anita@bmscw.edu.in

[2] Department of Computer applications, Arulmigu Arthanareeswarar Arts and Science college, Thiruchengodu, Tamilnadu, India; s.prema@ksrcas.edu.in

* Correspondence: anita@bmscw.edu.in; Tel.: +91 8197675637

Abstract: Forensic Science is the application of Scientific methods to resolve crime and legal issues. It involves various disciplines, such as Computer Science, Biology, Chemistry and Anthropology. Forensic scientists examine and analyze evidence from crime scenes, such as fingerprints, DNA, blood, or weapons. Digital proof is one of the forms of forensic evidence. It provide real time eye witness of the incident. Video recordings enable investigators to find out what exactly has transpired. Investigators use video evidence as a source for witness statements, and it aids in the search for the missing person or suspect. Video evidence is also used to testify in court and help with investigations and prosecutions. Failure of forensic science results in wrong judgement convicting innocent people and escaping criminals [1]. For most crimes high quality video recordings are often not available. video quality issues such as blurry, speckled, pixelated and low-resolution videos captured at low light are a real challenge in forensic analysis. To address such issues in this research a hybrid model using set of filters including triplemask spatial linear filter, median filter and bilateral filters are used. For denoising images, a novel image filter using sliding window convolution is proposed. For image sharpening a triplemask spatial linear filter is proposed. Triplemask spatial linear filter is created by cascading a series of filters. Identity, shift and fraction-based approach is used in mask processing. For image smoothing and to preserve the edges bilateral filter is used [2]. The performance of convolution operation is compared with distinct convolution, shift rotational convolution and scipy convolution. To handle uncertainty, imprecision, and ambiguity in real-world image data in a precise manner neutrosophic science is used in image analysis. By the generated neutrosophic set of the given input image ambiguous regions in the image are detected. Feature selection is made by calculating the entropy of different image regions. From the generated neutrosophic set entropy the degree of uncertainty, within the input image is quantified. The intensity distributions are measured using entropy values. In feature selection regions with highest and lowest entropy values containing face images are selected, visualized and processed to further aid in forensic analysis in detecting the culprits. Neutrosophic AHP is used for prioritizing criteria based on face detection and indeterminancy. Face detection is performed using single shot detector framework with a resnet base network, trained using caffe deep learning framework. face recognition process is performed using dlibs [9] state-of-the-art face recognition model built with deep learning [10]. Face recognition in this research distance-based similarity measure using neutrosophic sets is performed. These measures are used in conjunction with facenet[59] face recognition algorithm to improve the robustness and accuracy over traditional methods. The model has an accuracy of 99.38% on the labelled faces in the wild benchmark.

Keywords: Forensic; video-preprocessing; neutrosophic-sets; face-detection; face-recognition, deep learning.

1. Introduction

Forensic investigations depend on the quality and clarity of video evidence. However, several challenges are associated with video resolution in forensic applications such as low-Quality Source Footage, Digital Zoom Artifacts, Compression Artifacts, Limited Field of View, Frame Rate Issues, Interlacing, Challenges in Enhancement, Authentication and Tampering, Low-Light Conditions and Integration of Multiple Sources. To address these challenges, forensic experts often employ various techniques and tools, such as image and video enhancement software, upscaling algorithms, and specialized video analysis software. Additionally, advances in surveillance technology, including high-resolution cameras and improved compression algorithms, are gradually improving the quality of video evidence available for forensic purposes. In this paper, to address Low-Quality Source Footage and Low-Light Condition issues in Forensic video analysis Deep learning techniques and neutrosophic set theory are used. In studying the related works based on the selected research problem bibliometrics analysis is performed. In this paper vos-viewer tool is used to construct and visualize bibliometric networks based on the research problem. Dimensions database is used to study the related work based on the selected problem statement. The search analysis is made with the Co-authorship and authors relation in Full counting mode. The search query “crime AND Videoevidence AND forensics” is used. The query response displays only 9 publications from 2014 to 2023 in forensics, crime, and video evidence. The search analysis was performed with the threshold criteria: Minimum number of documents of an author: 1. Minimum number of citations of an author: 20. Out of 14 authors 6 meet the threshold.

Table 1. Author documents and citations

S.No.	Author	Documents	Citations	Total Link Strength
1	**Green, Sarah L**	1	24	2
2	**Mine, Becky**	1	23	2
3	**Niche, Robert A.**	1	24	2
4	**Powell, Martine B.**	1	27	1
5	**Wade, Kimberely A.**	1	24	2
6	**Westera, Nina J.**	1	27	1

After verifying the details of 6 selected authors a total-strength of Co-Authorship is calculated. The authors with the maximum link strength are selected. Some of the 6 items in the network are connected to each other. The network represents the Author and Co-Author relation between Kimberley A, Wade, Sarah. L, Green, Robert. A, Nash. The work carried by the authors is taken for reference in understanding how tampered video evidences induce false eye-witness and misleads in judgement.

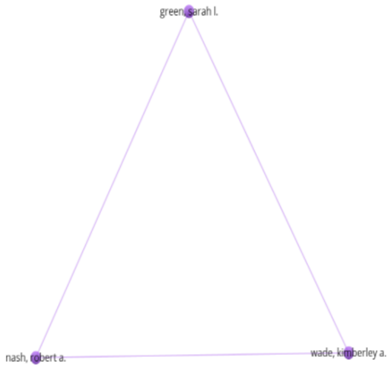


Figure 1. Bibliometric networks

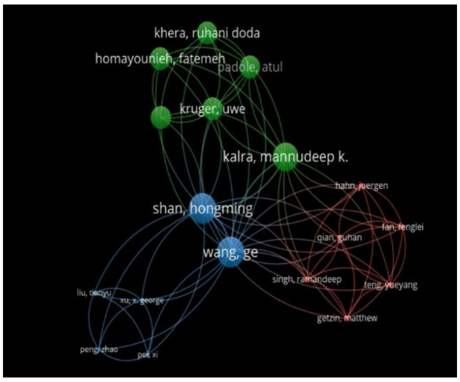


Figure 2. Network Visualization

The authors have proposed cognitive mechanisms in overcoming such defects in video analysis [21][22]. Performing Analysis with less amount of digital evidence in crime and violation prosecutions is a persistent problem. In Prosecutors' perceptions authors describe on improving the quality of evidence [23]. In "lost in the detail" paper authors discuss about prosecutors' perceptions of the utility of video recorded police interviews as rape complainant evidence. Strategy for improving police interviews in dealing with rape complaint cases to collect better evidences is discussed [24]. In amateur censoring visual information is described [25].

A neutrosophic set is a mathematical theory introduced by Florentin Smarandache in late 1990s as an extension of fuzzy sets [33]. Neutrosophic sets are used to represent and handle uncertainty, indeterminacy, and imprecision in a more comprehensive manner. A neutrosophic set consists of three components: the truth-membership function (T(x)), the indeterminacy-membership function (I(x)), and the falsity-membership function (F(x)). The sum of these membership values for any element x in the neutrosophic set should satisfy the inequality: $T(x) + I(x) + F(x) \leq 3$.

Neutrosophic science is an interdisciplinary field which collaborates with a diverse set of branches in scientific, engineering, computer science, mathematics, philosophy, and much more. Neutrosophic science has a wide range of applications including artificial intelligence, computer vision and pattern recognition [34,36], medical image processing, Segmentation [35] and Object Extraction, Video Analysis, Content-Based Image Retrieval, Image and Video Compression, Human-Computer Interaction, decision making, information fusion, philosophical implications and inter-disciplinary research. Recent researches using Neutrosophic Set in varying applications include the following: This study aims to analyse the cardiovascular disease risks by a new distance-based similarity measure motivated from intuitionistic fuzzy set theory[40]. They concluded that the existing measures failed in some circumstances, while the proposed measures classify them more appropriate and precisely. In this article the shortest route issue is formulated using an interval set of values in a Fermatean neutrosophic setting[39]. In this study the authors aim for an Optimal solution for Agricultural Land Use using an Efficient Neutrosophic Linear Programming Method[41]. This study highlights Innovations of Industry 5.0 Technologies Toward Smart Supply Chain Partners using AHP and single value neutrosophic sets (SVNSs)[42]. In this study Heart Disease Prediction under Machine Learning and Association Rules under Neutrosophic Environment is performed[43]. In image processing the input image is converted to digital data and it is processed to find more accurate information. In video processing, the input video is decomposed into a set of frames. Each frame is converted to an array containing details of hue, saturation and intensity of pixels. Pixel operations such as log, inverse log, inverse intensity, log, and gamma corrections [37] are performed on individual pixels to correct dead pixels, for denoising, smoothing and image quality enhancement.

2. Video Evidence based Forensic Analysis and its challenges

In the process of finding the real accused, recorded video evidences are used in Forensic analysis. For forensic analysis based on video evidences accurate visual information is essential for proper investigation and to make right judgment. A good quality video aids in proper investigation and clear analysis of crime cases. Certain video quality issues significantly hampers forensic analysis in following scenarios:

2.1. Surveillance video footage analysis

Low standard surveillance video footages affected by pixelation, blurriness and poor lighting, becomes an obstacle in identifying the individuals, vehicles and other objects of interest.

2.2. Crime scene remaking

Video footages of crime scenes are used as base reference for remaking the scene and to derive conclusions from visual information. Bad quality videos lead to ambiguity in finding certain minute details.

2.3. Facial detection and recognition

One of the applications of facial recognition technology is to detect and recognize faces. From poor quality videos affected with noise or blur it is difficult to accurately match faces to known individuals.

2.4. Vehicle license plate Identification

In cases of accidents video footages involving vehicle details are used in analysis. Poor video quality obscures readability of plates, making it challenging to identify vehicles involved in incidents.

2.5. Digital Forensics and Authentication

Preserving the integrity and quality of the original video footage is vital for authentication and creating a chain of custody.

2.6. Video enhancement and evidence analysis

Videos collected certain times requires enhancement to reveal hidden details. Bad quality video footages hinder the effectiveness of enhancement techniques leading to inaccurate interpretations.

2.7. Expert witness testimony

Forensic experts often depend on video evidences to support their testimony in court. Poor video quality weakens the reliability of expert opinions and deteriorate the influential impact of visual evidence on the judges.

2.8. Documenting human rights violations

In cases involving human rights abuses and criminal activities, video recordings are used as vital evidence. If the quality of such recordings is not to the standard, it can hamper the efforts in documentation and impeach criminals.

2.9. Accident Reconstruction

Video footage of accidents captured by traffic cameras and surveillance cameras, is important for remaking events and determining accountability. Poor video quality obscures important details for accurate reconstruction.

3. Proposed Methodology

Spatial Linear filter is chosen over spatial non-linear filter because of its computational efficiency, simplicity and interpretability, linearity preserving nature, preserves structural information, imposes smoothness and denoising, and compatible with convolutional neural network.

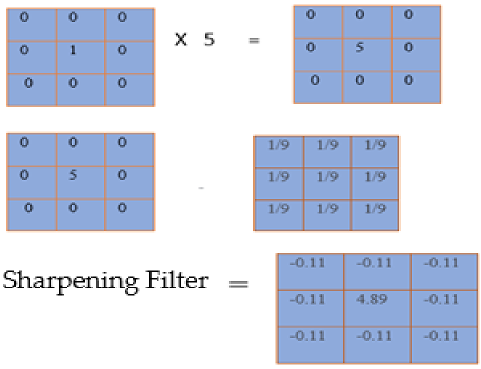
The computational complexity of linear filters is based only on the size of the filter/kernel which has reduced computations. Spatial non-linear filters involve complex sorting and iterative processes which makes computationally inefficient. Spatial linear filters are known for its simplicity and interpretability. Linear filters are easy to understand and analyze. To detect faces in videos involving various linear transformations and interpret the image using neutrosophic set mathematical operations linear filter is used. Spatial linear filter is chosen for preserving the structural details of the image. Linear filters are selected to impose smoothness over the output image by denoising it. Linear filters are compatible with convolutional neural networks. To this end the convolution operations are directly imposed at convolutional layers for face detection.

3.1. TripleMask Spatial Linear filter

Convolution operation is performed on the selected input image. Convolution operation on an image is a process of adding each element of the image to its local neighbors, weighted by a small matrix called the kernel [26]. The kernel acts as a filter that modifies the output image according to some desired effect, such as blurring, sharpening, edge detection, etc. [27]. The properties of convolution, such as commutativity, associativity and distributivity, still hold for image convolution [28]. Convolution properties are as follows:

1. It is commutative: $a * b = b * a$
2. It is distributive with respect to addition: $a * (b1 + b2) = a * b1 + a * b$
3. It is linear: $t\ (a * b) = (t \cdot a) * b$
4. It is separable if the kernel is separable: $a * (bi \otimes bj) = (a * bi) * bj$

A Triple Spatial Linear filter is proposed for image sharpening. The filter is created by cascading a series of filters. Identity, Shift and Fraction based approach is used in selecting the Mask. A 64x64 size image is taken as input. The input image is processed by applying spatial linear filter. Spatial linear filter is created by choosing an identity kernel and a fractional kernel. To increase the strength of kernel values by using the properties of convolution, the identity kernel elements are multiplied by an integer. The resultant product kernel is then subtracted with fractional kernel. By doing this we have obtained a simple sharpening filter. For Image smoothing and to preserve the edges bilateral filter is used [2].



$$\begin{bmatrix}0&0&0\\0&1&0\\0&0&0\end{bmatrix} \times 5 = \begin{bmatrix}0&0&0\\0&5&0\\0&0&0\end{bmatrix}$$

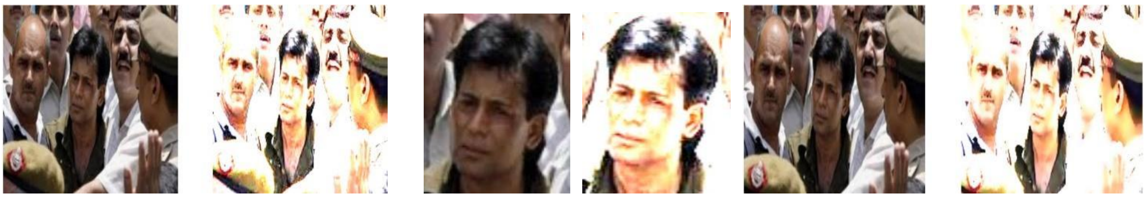
$$\begin{bmatrix}0&0&0\\0&5&0\\0&0&0\end{bmatrix} - \begin{bmatrix}1/9&1/9&1/9\\1/9&1/9&1/9\\1/9&1/9&1/9\end{bmatrix}$$

$$\text{Sharpening Filter} = \begin{bmatrix}-0.11&-0.11&-0.11\\-0.11&4.89&-0.11\\-0.11&-0.11&-0.11\end{bmatrix}$$

Figure 3: Sharpening Filter

Triple Spatial Linear filter procedure:

```
Convert input image as array
Arrayimage = asarray(image)
Declare identity kernel of size m x n
//assign kernel elements
Identity kernel = [[0] * n for i in range (m)]
Display identity kernel
Declare fractionalkernel of size m x n
//assign kernel elements
Fractionalkernel[m][n] = [[1/9] * n for i in Range (m)]
Display fractionalkernel
Set constant = k
Multiple identity kernel elements by constant k
Product identity kernel[i][j] = identity kernel * k
Obtain new sharpeningfilter
Sharpeningfilter[i][j] = productidentitykernel[i][j] - fractionalkernel[m][n]
Perform convolution operation
Rotate Sharpeningfilter by 180 degree
//apply rotatedsharpeningfilter on input image
Sharpenedimage=filter2d(arrayimage, ddepth = -1, kernel= rotatedsharpeningfilter)
Display sharpenedimage
```



(a) (b) (c) (d)

Figure 4. Input image processed using sharpening filter and bilateral filter is listed as: (a) Original image and sharpen image; (b) Cropped original; (c) sharpened image; (d) Image smoothing and edge preserving.

3.2. Advantages of using proposed TripleMask spatial linear filter

3.2.1. Improved feature representation

The convolution operation using identity filter on image preserves the original feature, the shift operation on the image introduces translational invariance and fractional mask/kernel operation enables precise fine-grained feature extraction for precise localization to detect and segment faces in a video and image. By combining these three kernels/filters we obtain more comprehensive image feature representations.

3.2.2. Enhanced robustness to basic geometric transformations

Shift and fractional kernel operations on input image makes the working model more robust to translation, rotation, scaling and distortion.

3.2.3. Reduction in overfitting

By combining three filters identity, shift and fraction kernels the model is made to learn diverse representations of data to reduce overfitting when trained with limited data.

3.2.4. Improved level of performance

By using the combined filter approach the model can leverage the strengths of each kernel operation to result in improved performance in face detection and segmentation.

3.2.5. Adaptability

The kernel operation is refined using distributive, commutative and linear convolution properties to fit to the model architecture based on the input dataset.

3.3. Convolution Operations, Impact on Video Quality Enhancement

Different convolution operations excel in various scenarios based on their characteristics and the requirements of the task at hand. Here we write few insights on each convolution operation, their influence to video quality enhancement and the scenarios where they excel:

3.3.1. Mean Filter:

Scenario: Mean filtering is operative for reducing Gaussian noise in images or videos. It is mainly useful in scenarios where the noise is uniform and can be modelled as Additive-white gaussian noise.

Contribution to Video Quality Enhancement: By smoothing noise, the mean filter helps improve the visual quality of videos, making them appear cleaner and less grainy.

3.3.2. Gaussian Filter:

Scenario: Gaussian filtering is appropriate for noise reduction while preserving edges and fine details in the image or video. It is commonly used when a more natural and visually pleasing result is looked-for compared to traditional mean filtering.

Contribution to Video Quality Enhancement: Gaussian filtering effectively reduces noise while maintaining the sharpness of edges and textures, resulting in visually attractive videos with improved clarity.

3.3.3. Sobel and Prewitt Operators:

Scenario: Sobel and Prewitt operators are commonly used detecting edges in images and videos. They are effective in detecting regions of rapid intensity changes, and object-boundaries. *Contribution to Video Quality Enhancement:* These operators are used in enhancing the perceptual quality of videos by highlighting the edges. They advance in the delineation of objects and structures within the video frames, resulting in much sharper and well-defined visual content.

3.3.4. Laplacian Filter:

Scenario: The Laplacian filter is useful for edge detection and image sharpening. It highlights regions of rapid intensity changes and enhances the overall contrast in the image or video.

Contribution to Video Quality Enhancement: By highlighting edges and fine details the Laplacian filter enhances the visual quality of videos by making them appear much sharper and more detailed.

3.3.5. Gabor Filter:

Scenario: Gabor filters are commonly used for texture analysis and feature extraction in images and videos. They are effective in capturing both spatial and frequency information by making them suitable for tasks such as texture classification and segmentation.

Contribution to Video Quality Enhancement: Gabor filters is used to improve the perceptual quality of videos by capturing and preserving texture details. They enhance the richness and depth of textures by leading to more visually pleasing videos.

3.3.6. Wiener Filter:

Scenario: The Wiener filter is an adaptive filter used for noise reduction in images and videos. It estimates the power spectrum of the noise and the signal to compute an optimal filter that minimizes the mean square error.

Contribution to Video Quality Enhancement: By adaptively reducing noise while preserving image details the Wiener filter significantly enhances the visual quality of videos. It ensures that noise reduction is optimized for the specific characteristics of the noise and the underlying signal by resulting in clear and high-quality video content.

3.4. Performance Comparison of convolution operation

The performance of convolution operation is compared with a distinct convolution, a shift rotational convolution and scipy convolution. The performance is measured using various metrics in terms of CPU times: user time and sys time, Wall time and Total time. Scipy convolution operation comparatively is efficient in which user time- the amount of CPU time taken outside of the kernel and the sys time- the amount of time taken inside of the kernel is very least amount measuring only to 2 µs and 1e+03 ns where as traditional Convolution operation takes user time: 208 ms and sys time: 50.1 ms. Comparing Sliding window convolution with Distinct and traditional convolution the CPU times its measures only 10% of 305 ms to 14% of 208 ms and Wall time measures 11% of 311 ms to 12% of 294 ms.

3.4.1. Traditional Convolution

$$Z(i, j) = w * f(i, j) = \sum_{di=-a}^{a} \sum_{dj=-b}^{b} w(di, dj)\, f(i-di)(j-dj) \quad (1)$$

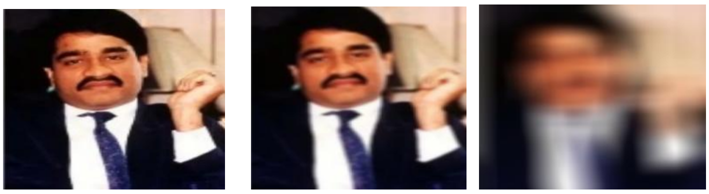
Z(i,j) is the filtered image, f(i,j) is the original image, w is the filter kernel. Where $-a \le di \le a$ and $-b \le dj \le b$.

3.4.1.1. Image smoothening

Performing convolution operation on an input image with, selected kernel of Ones is shown as follows:

Kernel= Ones ((size, size)) / (size ^2) (2)

size=3, kernel equation (2) generates a kernel as follows: [[1. 1. 1.], [1. 1. 1.], [1. 1. 1.]]



(a) (b) (c)

Figure 5. Traditional Convolution operation for Image smoothening with kernel in equation 2 using mean kernel smoothening produces results with varying size listed as: (a) Original image; (b) 3x3 kernel; (c) 20x20 kernel. (CPU times: user 208 ms, sys: 50.1 ms, total: 258 ms, Wall time: 294 ms)

3.4.2. Distinct Convolution

$$Z(i, j) = M * f(i, j) = \sum_{di,\, dj} f(i+di, j+dj) * M[di, dj] \quad (3)$$

Z(i,j) is the filtered image, f(i,j) is the original image, M is the filter kernel. Where -a≤di≤a and -b≤dj≤b. Performing convolution operation on an input image with, selected kernel of Ones is shown as follows:

$$\text{Kernel} = \text{ones}((21,3))/(21/3) \quad (4)$$

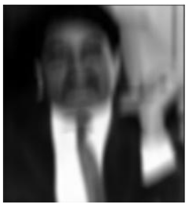


Figure 6. Distinct convolution operation (CPU times: user 305 ms, sys: 0 ns, total: 305 ms, Wall time: 311 ms)

3.4.3. Sliding window convolution

A sliding window which highlights the image pixels to be operated is covered by the window. Image values are not copied but viewed. It is faster with the sliding window since there is no loop. The creation of the rolling window does not take much time because no data is copied, it is just a view on the original matrix. Performing sliding window convolution operation on an input image with selected kernel ones using equation (4) is shown as follows:

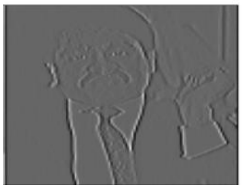


Figure 7. Sliding window convolution operation

CPU times: user 31.5 ms, sys: 6.34 ms, total: 37.9 ms, Wall time: 36.9 ms

3.4.4. Scipy convolution

Scipy convolution operation is performed using functions linear and non-linear filtering, binary morphology, B-spline interpolation [29].

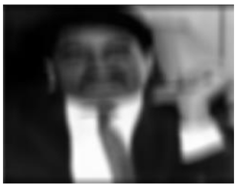


Figure 8. Scipy convolution operation

CPU times: user 2 μs, sys: 1e+03 ns, total: 3 μs, Wall time: 5.96 μs

Table 2. Performance Comparison of convolution operation

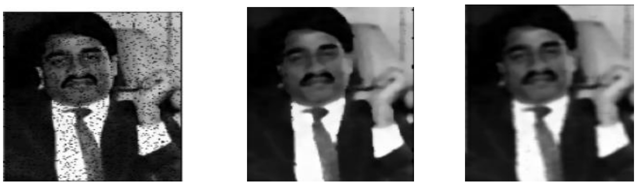
S.no	Operation	CPU times		Wall time	Total CPU times
		user	sys		
1	Traditional Convolution	208 ms	50.1 ms	294 ms	258 ms
2	Distinct Convolution	50.1 ms	0 ns	311 ms	305 ms

3	Sliding Window Convolution	305 ms	6.34 ms	36.9 ms	37.9 ms
4	Scipy Convolution	2 µs	1e+03 ns	5.96 µs	3 µs

Scipy convolution operation comparatively is efficient in which (user time) the amount of CPU time taken outside of the kernel and (sys time) the amount of time taken inside of the kernel is very least amount measuring only to 2 µs and 1e+03 ns where as Traditional Convolution operation takes user time: 208 ms and sys time: 50.1 ms. Comparing Sliding window convolution with Distinct and traditional convolution the CPU times its measures only 10% of 305 ms to 14% of 208 ms and Wall time measures 11% of 311 ms to 12% of 294 ms.

3.5. Image Denoising

To denoise the input image sliding Window Convolution operation is used. Creating a new novel filter by using sliding window convolution to denoise images is shown below. For each sliding window sort the pixels and take the mean(r) of the pixel. When r=1 the filter functions similar to median filter. If r=3 then novel filter function is invoked. An image affected with impulse noise is taken as input. Its is processed using novel filter. The output is compared with median filter. The obtained result is 99% similar to that of median filter.



(a) (b) (c)

Figure 9. Denoising input image using sliding Window Convolution operation produces results as listed (a) Original image; (b) Median filter; (c) Novel filter.

3.6. Neutrosophic Science in Forensic Analysis: Addressing Uncertainty and Ambiguity, Image Data

Neutrosophic science has several advantages in addressing uncertainty and ambiguity in image data over forensic analysis:

1. *Handling uncertainty in Video Evidence Interpretation*: Forensic analysis often deals with imperfect or incomplete evidence leading to uncertainty in interpretation. Neutrosophic logic provides a formal framework to represent and reason about uncertainty in image data. By assigning truth-membership, indeterminacy membership, and falsehood membership degrees to pixel values or image features. Neutrosophic science allows forensic analysts to model and quantify uncertainty more effectively.
2. *Robustness to Noise and Distortion:* Images obtained as evidence in forensic investigations may be subjected to noise, compression artifacts, or other forms of distortion. Neutrosophic image processing techniques, such as neutrosophic filtering and denoising, are designed to handle such imperfections while preserving important details. By incorporating neutrosophic set theory into image processing algorithms, forensic

analysts can enhance the quality of image evidence and reduce the impact of noise and distortion on analysis results.

3. *Accurate Boundary Detection and Segmentation:* Neutrosophic edge detection and segmentation methods are robust to ambiguity in object boundaries, making them well suited for forensic applications where precise delineation of objects or regions of interest is crucial. By considering indeterminacy membership degrees in addition to truth membership degrees, Neutrosophic segmentation algorithms can accurately segment forensic images even in cases where object boundaries are unclear or ambiguous.
4. *Effective Fusion of Multimodal Data:* Forensic investigations often involve the analysis of multiple types of evidence, including images, videos, and other sensor data. Neutrosophic image fusion techniques enable the integration of information from diverse sources while accounting for uncertainties inherent in each modality. By combining neutrosophic set theory with fusion algorithms, forensic analysts can extract more comprehensive and reliable information from multimodal data thus enhancing the overall accuracy and completeness of forensic analyses.
5. *Transparent Representation of Evidence Confidence:* Neutrosophic science provides a transparent way to represent the confidence level or reliability of evidence in forensic analyses. By explicitly quantifying truth membership, indeterminacy membership, and falsehood membership degrees. Neutrosophic logic allows forensic analysts to convey the degree of uncertainty associated with different aspects of image data. This transparency fosters better communication and interpretation of forensic findings.

4. Preliminaries

This section introduced some preliminary notions which will be applied in the final analysis.

4.1. Single Valued Neutrosophic Set

A neutrosophic set which can be used in real scientific and engineering applications is known as Single valued neutrosophic set (SVNS).

Definition 4.1.1 [46]. Let X be a space of points (objects) with a generic element in X denoted by x. A single valued neutrosophic set A in X is characterized by a truth membership function, $TA(x)$, an indeterminacy membership function, $IA(x)$, and a falsity membership function $FA(x)$. Here $TA\ (x)$, $IA\ (x)$, $FA\ (x)$ are real subsets of [0,1].

$$A = \{\langle x,\ TA\ (x), IA\ (x),\ FA\ (x)\rangle \mid x \in X\}$$

4.2. Distance-based Similarity Measure of Neutrosophic set

Definition 4.2.1 [47]. Normalized Hamming distance measure $dNS\ NH(A,\ B)$ operator between neutrosophic set A and B is defined as follows:

$$d_{NS}^{NH}(A,B) = \frac{1}{3n}\sum_{i=1}^{n}\left(|T_A(x_i) - T_B(x_i)| + |I_A(x_i) - I_B(x_i)| + |F_A(x_i) - F_B(x_i)|\right)$$

Definition 4.2.2 [47]. Normalized Euclidean distance measure $dNS\ NE(A,\ B)$ operator between neutrosophic set A and B is defined as follows:

$$d_{NS}^{NE}(A,B) = \sqrt{\frac{1}{3n}\sum_{i=1}^{n}\left((T_A(x_i) - T_B(x_i))^2 + (I_A(x_i) - I_B(x_i))^2 + (F_A(x_i) - F_B(x_i))^2\right)}$$

Definition 4.2.3 [47]. Let A, B be two neutrosophic sets in X. The similarity measure between the neutrosophic sets A and B can be evaluate from distance measures, as follows:

$$S_N(A,\ B) = 1 - d_{NS}(A,\ B)$$

where $d_{NS}(A,\ B)$ is represent the distance measure between neutrosophic set A and B for all $xi \in X$.

Proposition 4.2.3.1: The distance measures for neutrosophic set $d_{NS}(A,\ B)$and similarity measure for neutrosophic set $S_N(A,\ B)$ satisfies the following properties:

$0 \leq d_{NS(}A,\ B) \leq 1$; $0 \leq S_N(A,\ B) \leq 1$;

$d_{NS}\ (A,\ B) = 0$ if and only if A = B; $S_N\ (A,\ B) = 1$ if and only if for $A = B$

$d_{NS}\ (A,\ B) = d_{NS}\ (B,\ A)$; $S_N\ (A,\ B) = S_N\ (B,\ A)$;

$d_{NS}\ (A,\ C) \leq d_{NS}\ (A,\ B)$ and $d_{NS}\ A,\ C) \leq d_{NS}\ (B,\ C)$ if C is neutrosophic set in X and $A \subseteq B \subseteq C$;

$S_N\ (A,\ C) \leq S_N\ (A,\ B)$ and $S_N\ (A,\ C) \leq S_N\ (B,\ C)$ if C is neutrosophic set in X and $A \subseteq B \subseteq C$. [44,45]

5. Represent Image In Neutrosophic Set

Representing an image in neutrosophic set theory involves characterizing the uncertainty and indeterminacy present in the image's pixel values. Neutrosophic set theory is an extension of fuzzy set theory that introduces a third membership function, representing the degree of indeterminacy. First the input image is read and preprocessed to convert to a format suitable for applying neutrosophic science techniques. The image is loaded using Keras with parameters colormode rgb and interpolation nearest. The input image is converted to a tensor using tensorflow. The shape of the tensor is determined using tensorflow. The input image has a tensor shape of 194 X 259 with 3 colour channels [194, 259,3], with its data elements as int32 type. The created tensor is converted to a NumPy array.

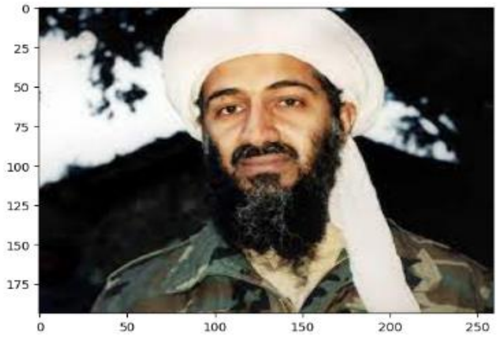


Figure 10. Input image

[[13. 8. 14.] [11. 6. 12.] [4. 0. 5.]	[6. 6. 6.] [6. 6. 6.] [6. 6. 6.]] ...	[[3. 0. 4.] [1. 0. 4.] [3. 0. 4.]	[6. 6. 6.] [6. 6. 6.] [6. 6. 6.]] …	[[1. 0. 4.] [0. 0. 4.] [1. 0. 4.]	[6. 6. 6.] [6. 6. 6.] [6. 6. 6.]]

Figure 1l. Image to Array

Generate the Neutrosophic Set Elements for the image_array created above. Representing an image as a neutrosophic set in Python involves assigning each pixel in the image degrees of membership in the T (true), I (indeterminate), and F (false) subsets as per the definition in 4.1.1 [46]. For representing in neutrosophic set domain the degrees of membership for each pixel is defined using two membership rules:

Rule 1: If intensity > 128, assign high T-membership, low I-membership, and low F-membership. In first rule T-membership value is normalized to [0, 1], having T_membership = (pixel - 128) / 127.0, I_membership = 0.2 and F_membership = 0.1

Rule 2: If intensity <= 128, assign high F-membership, low I-membership, and low T-membership. In second rule F-membership value is normalized to [0,1], having T_membership = 0.1, I_membership = 0.2, F_membership = (128 - pixel) / 127.0.

The image is converted to gray image and the intensity of each pixel is read from pixel coordinates x,y. It is observed that pixels with 0 intensity are ambiguous and unclear.

```
Pixel at (0, 175) has intensity: 0
Pixel at (1, 175) has intensity: 0
Pixel at (2, 175) has intensity: 0
Pixel at (3, 175) has intensity: 0
Pixel at (4, 175) has intensity: 0
Pixel at (5, 175) has intensity: 0
Pixel at (6, 175) has intensity: 0
Pixel at (7, 175) has intensity: 0
```

Figure 12. Ambiguous and unclear

```
[0.08661418, 0.11811024, 0.33070865]
[0.00000000, 0.00000000, 0.22047244]
[0.18897638, 0.17322835, 0.33858266]
[0.24409449, 0.24409449, 0.43307087]
```

Figure 13. (F,I,T)-membership for pixels in figure 12

The pixels displayed in figure:12 consists of regions which are difficult to interpret. Recognizing such regions are crucial for image understanding and decision-making. For these pixels neutrosophic set contains high F-membership, low I-membership, and low T-membership.

```
Pixel at (134, 181) has intensity: 128
Pixel at (135, 181) has intensity: 107
Pixel at (136, 181) has intensity: 143
Pixel at (137, 181) has intensity: 164
Pixel at (138, 181) has intensity: 190
Pixel at (139, 181) has intensity: 207
Pixel at (140, 181) has intensity: 211
Pixel at (141, 181) has intensity: 210
Pixel at (142, 181) has intensity: 211
```

```
[ 0.42519686, 0.26771653, 0.33858266],
[ 0.44094488, 0.28346458, 0.35433072],
[ 0.40157488, 0.24409449, 0.31496063],
[ 0.40944883, 0.25196858, 0.32283464],
[ 0.44881899, 0.31496063, 0.37795275],
[ 0.41732284, 0.28346458, 0.34645677],
[ 0.47244096, 0.33858266, 0.40157489],
[ 0.62992126, 0.52755904, 0.58267724]
```

Figure 14. Pixels with intensity > 128: unambiguous and clear **Figure 17.** High T-membership, low I-membership, and low F-membership for pixels in figure 14

5.1. Pixel-wise Neutrosophic Entropy Calculation

Neutrosophic-entropy has wide significance in image processing [38]. Neutrosophic set entropy is a measure of the uncertainty. To calculate entropy for input image, the input_arr equivalaent for the input image is converted to float type value using the operation input_arr.astype(float)/255.0. Calculate neutrosophic entropy for each pixel having membership > 0, using the following equation:

entropy -= membership * np.log(membership + epsilon) (5)

Where, epsilon = 1e-10 and pixel_memberships = [0.3, 0.4, 0.3].

```
[[[0.15173366 0.10860618 0.15933681]
  [0.13559628 0.08822363 0.14382856]
  [0.06517599 0.         0.07709462]
  ...
  [0.08822363 0.08822363 0.08822363]
  [0.08822363 0.08822363 0.08822363]
  [0.08822363 0.08822363 0.08822363]]
```

```
 [[0.05226649 0.         0.06517599]
  [0.02173045 0.         0.06517599]
  [0.05226649 0.         0.06517599]
  ...
  [0.08822363 0.08822363 0.08822363]
  [0.08822363 0.08822363 0.08822363]
  [0.08822363 0.08822363 0.08822363]]
```

Figure 12. Neutrosophic entropy of input image in figure 10

It is observed for those pixels having intensity values with equal probability, the entropy is maximum, and for pixels having low intensity the entropy values are low indicating nonuniform intensity distribution. Pixel at (10,60) has intensity: 2 resulting with low entropy ((10, 60), 0.8812909).

5.2. Identification of Crucial Image Regions from Calculated Entropy Values

The input image shape is read and stored as "height, width = gray_image.shape", for each pixel coordinates x,y in for y in range(height) and for x in range(width), intensity is found as "f'Pixel at ({x}, {y}) has intensity: {intensity}'". The input image is read in rgb color mode with 'nearest' interpolation, convert to input image to array and degrees of membership for each pixel is defined, for each pixel in row: If (pixel intensity > 128).any() then t_membership = (pixel - 128) / 127.0, normalize T-membership to [0, 1], i_membership = 0.2, f_membership = 0.1. Else: t_membership = 0.1, i_membership = 0.2, f_membership = (128 - pixel) / 127.0, normalize F-membership to [0, 1]. Append the degrees of membership for each pixel, row_set.append((t_membership, i_membership, f_membership)).

Thresholding is applied to identify indeterminate regions. An indeterminacy_filter is defined to work the input image with threshold =100. Indeterminacy_filter(image, threshold=100). _, binary_image = cv2.threshold(image, threshold, 255, cv2.THRESH_BINARY). Then the binary image is inverted to get indeterminate regions. Indeterminate_regions = cv2.bitwise_not(binary_image).

Neutrosophic_entropy is calculated for pixel_memberships >0, initialy entropy is set to be 0.0 and a small constant epsilon, to avoid taking the logarithm of zero is used, where epsilon = 1e-10. for membership in pixel_memberships, if membership > 0 then entropy is calculated as entropy -= membership * np.log(membership + epsilon). Example degrees of membership for a pixel (T, I, F) as pixel_memberships = [0.3, 0.4, 0.3], Neutrosophic entropy for the pixel is calculated as Neutrosophic Entropy: 1.0888999750452237. Calculating the neutrosophic entropy for the entire input image, entropy = neutrosophic_entropy(image_float), resulting as: Neutrosophic Entropy: [[[0.15173366 0.10860618 0.15933681], [0.13559628 0.08822363 0.14382856], [0.06517599 0. 0.07709462] ... [0.08822363 0.08822363 0.08822363], [0.08822363 0.08822363 0.08822363].

It is observed for those pixels having intensity values with equal probability, the entropy is maximum, and for pixels having low intensity the entropy values are low indicating nonuniform intensity distribution. Pixel at (10,60) has intensity: 2 resulting with low entropy ((10, 60), 0.8812909).

From the calculated entropy values, it is observed, when pixels having low intensity its entropy values are also low. Such pixels with low entropy values has nonuniform intensity distribution in the input image. Such finding helps in finding crucial image regions.

5.3. Entropy a Suitable Measure for Feature Selection in Forensic Face Feature Detection

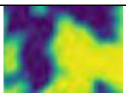
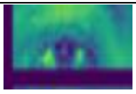
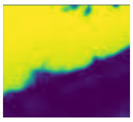
Entropy is a suitable measure for feature selection in this context due to several reasons. In handling uncertainty in image data, such as noise, occlusion, and variations in illumination and pose. Faces in images may exhibit varying degrees of uncertainty due to factors like partial occlusion or changes in lighting conditions. Neutrosophic entropy considers truth-membership, indeterminacy-membership, and falsehood-membership degrees, allowing it to capture uncertainty in feature selection more comprehensively compared to traditional entropy measures. Neutrosophic Entropy is robustness to Ambiguity in feature selection. Faces in images may appear differently due to variations in facial expressions, age, gender, and ethnicity, leading to ambiguity in feature representation. Neutrosophic entropy considers indeterminacy-membership degrees, which reflect the degree of ambiguity associated with feature selection. Neutrosophic entropy can integrate information from multiple modalities for feature selection. In face detection, different modalities such as color, texture, and shape may contain complementary information about facial features. Neutrosophic entropy allows for the fusion of information from these modalities while considering their respective truth-membership and indeterminacy-membership degrees, enabling more comprehensive feature selection.

The functionality of feature selection process using entropy for forensic face feature detection is described below. Load the image in grayscale, a function is defined to calculate entropy as follows: def calculate_entropy(image_patch): calculate histogram, hist = cv2.calcHist([image_patch], [0], None, [256], [0, 256]), normalize histogram, hist = hist / hist.sum(), calculate entropy for image patches, entropy = -np.sum(hist * np.log2(hist + np.finfo(float).eps)). Define the size of the image regions or patches, patch_size = 64, Iterate over the image and calculate entropy for each patch as follows:

```
for y in range(0, image.shape[0] - patch_size + 1, patch_size):
    for x in range(0, image.shape[1] - patch_size + 1, patch_size):
        patch = image[y:y+patch_size, x:x+patch_size]
        entropy = calculate_entropy(patch)
        patch_coordinates.append((x, y))
        patch_entropies.append(entropy)
```

Sort patches by entropy, sorted_patches = sorted(zip(patch_coordinates, patch_entropies), key=lambda x: x[1]), Select the patches with the highest entropy (or lowest, based on your needs), num_selected_patches = 10 # Adjust this value as needed, selected_patches = sorted_patches[-num_selected_patches:], Visualize or process the selected patches as needed, for (x, y), entropy in selected_patches: patch = image[y:y+patch_size, x:x+patch_size], cv2.rectangle(image, (x, y), (x+patch_size, y+patch_size), (0, 255, 0), 2) by drawing a rectangle around selected patches.

Table 3. Observation from calculated entropy values

Input	Parameters	High Entropy Features and values			Low Entropy Features and values		
Image	Degree of uncertainty	Greater uncertainty	((40, 30), 4.8966913)		More certainty	((130, 50), 5.8985624)	
Image	Feature selection	Important, variable information	((130, 70), 5.72637)		More constant color	((220, 110), -0.0)	
Image	Information content	More diverse, unpredictable information	((70, 80), 4.902795)		Constant, predictable information	((130, 50), 5.8985624),	

The entropy values calculated for patch coordinates for the figure: highlighted in red, has negative entropy values, with constant color, represents a kind of artifact or anomaly present in the image patch due to sensor noise or image corruption. The entropy values highlighted in blue, has high entropy values for the selected coordinates, signifies greater uncertainty in the image. The entropy values highlighted in light blue has low entropy values, representing more certainty facial feature an eyebrow. The entropy values highlighted in light green has high entropy values, representing more important, variable information such as mouth, mustache, part of nose, beard and ear tip.

5.3.1. Feature Selection based on Entropy values

Feature selection in image processing using neutrosophic sets is an approach that combines neutrosophic set theory with traditional feature selection techniques [39] to extract relevant information from images. First the input image is loaded. The function to calculate entropy is defined. the size of the image region is defined. To store region coordinates and entropy values a list is declared and initialized. The entropy is computed for each region of interest by iterating over the image. The entropy is calculated using equation (6). Histogram for the image_regions are calculated and normalized. The regions are sorted based on entropy values in ascending and descending order. The sorted regions are printed in the format of (region_coordinates, region_entropies). Upon selected regions a rectangle box is drawn to highlight the region.

$$\text{Entropy} = \text{sum}(\text{histogram} * \text{log2}(\text{histogram}) \tag{5}$$

```
    [((200, 100), -0.0), ((220, 110), -0.0), ((230, 110), -0.0), ((240, 140), -0.0), ((200, 100), -0.0), ((220,
  110), -0.0), ((230, 110), -0.0), ((240, 140), -0.0), ((240, 50), 0.22194068), ((240, 50), 0.22194068), ((210,
  110), 0.5293609), ((210, 110), 0.5293609), ((200, 110), 0.72192806),((110, 50), 6.121209), ((110, 80),
  6.121209), ((200, 80), 6.1412086), ((200, 80), 6.1412086), ((130, 80), 6.1536603), ((130, 80), 6.1536603),
  ((140, 50), 6.156307), ((140, 50), 6.156307), ((170, 70), 6.161209), ((170, 70), 6.161209), ((210, 80),
  6.2163067), ((210, 80), 6.2163067), ((10, 100), 6.308758), ((10, 100), 6.308758), ((170, 80), 6.3763065),
  ((170, 80), 6.3763065)]
```

Figure 13. Patch_coordinates and patch_entropies generated for input image in figure 14.

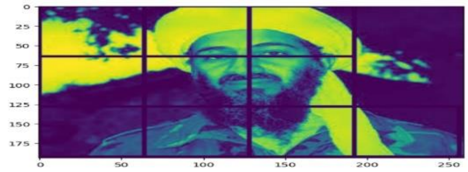
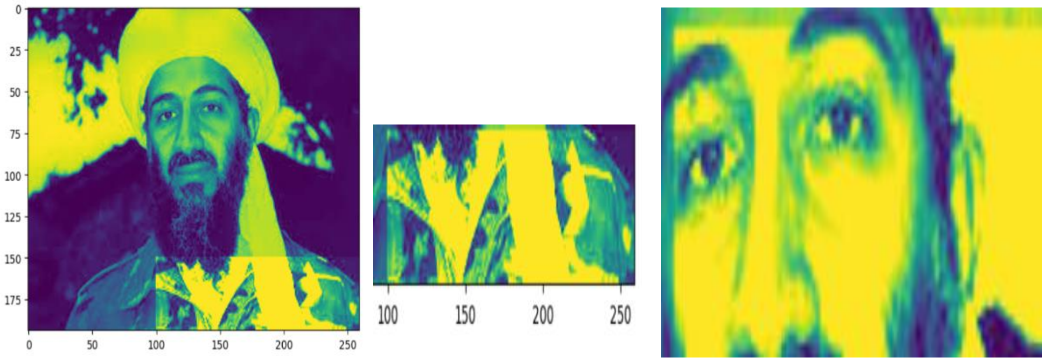


Figure 14. Patch_coordinates for input image

5.3.2. Region-Of-Interest (Roi) Enhancement Using Image Entropy

Region-of-Interest (ROI) enhancement involves applying image enhancement techniques to a specific region within an image. In this paper ROI enhancement is performed using OpenCV library. To enhance the region of interest selected first the image is loaded. The the coordinates (x, y) of the region to be enhanced is defined with its height and width factors as following: x, y, width, height = 100, 150, 200, 200. Using array slicing the selected region of interest is extracted. Then the enhancement technique is applied to the ROI. To enhance the ROI in this work colour contrast is selected. The colour contrast is increased using convertScaleAbs property of cv2.



(a) (b) (c)

Figure 15. Region-of-Interest enhancement using image entropy produces result as listed: (a) Increased color contrast for the region of interest: x, y, width, height = 100, 150, 200, 200; (b) Extracted region of interest using arrayslicing; (c) Color contrast increased using convertScaleAbs for the region of interest coordinates x, y, width, height = 100, 50, 200, 200

6. Distance-Based Similarity Measure For Face Recognition Using Neutrosophic Sets

In this paper for finding similarity between faces for face recognition in forensics investigation we use distance-based similarity measures using neutrosophic sets. Neutrosophic sets handle three types of indeterminacy: truth, indeterminacy, and falsity. Three types of distance-based similarity measures are used for face recognition using neutrosophic sets, such as Neutrosophic Euclidean Distance, Neutrosophic Cosine Similarity, Neutrosophic Manhattan Distance. The computations of these distances are given below.

6.1. Neutrosophic Euclidean Distance

Given two neutrosophic feature vectors A= (at,ai,af) and B= (bt,bi,bf). where at,ai,af and bt,bi,bf represent the truth, indeterminacy, and falsity degrees of feature A, and similarly for B, the neutrosophic Euclidean distance is computed as:

$$d(a,b) = \sqrt{(a_t - b_t)^2 + (a_i - b_i)^2 + \left(a_f - b_f\right)^2}$$

6.2. Neutrosophic Cosine Similarity

Given two neutrosophic feature vectors A and B, the neutrosophic cosine similarity is compu333ted as:

$$cos(a,b) = \frac{a_t \cdot b_t + a_i \cdot b_i + a_f \cdot b_f}{\sqrt{a_t^2 + a_i^2 + a_f^2}.\sqrt{b_t^2 + b_i^2 + b_f^2}}$$

6.3. Neutrosophic Manhattan Distance

Given two neutrosophic feature vectors A and B, the neutrosophic Manhattan distance is computed as:

$$d(a,b) = |a_t - b_t| + \quad |a_i - b_i| + |a_f - b_f|$$

6.4. Example:

Suppose we have the following neutrosophic feature vectors: A=(0.8,0.1,0.1), B=(0.6,0.2,0.2) , Here, A has a truth degree of 0.8, an indeterminacy degree of 0.1, and a falsity degree of 0.1. Similarly, B has a truth degree of 0.6, an indeterminacy degree of 0.2, and a falsity degree of 0.2.

Calculate the neutrosophic Euclidean distance between A and B using equation given in 6.1.

$$d(a,b) = \sqrt{(0.8-0.6)^2 + (0.1-0.2)^2 + (0.1-0.2)^2}$$

$$d(A,B) = \sqrt{0.06}$$

d(A,B) ≈ 0.24494897427831783

Calculate the neutrosophic cosine similarity between A and B using equation given in 6.2.

$$cos(a,b) = \frac{0.8 \cdot 0.6 + 0.1 \cdot 0.2 + 0.1 \cdot 0.2}{\sqrt{0.64 + 0.1 + 0.1}.\sqrt{0.36 + 0.4 + 0.4}}$$

$$cos(a,b) = \frac{0.52}{\sqrt{0.66}.\sqrt{0.44}}$$

$$cos(a,b) = \frac{0.52}{\sqrt{0.2904}}$$

$$cos(a,b) \approx \frac{0.52}{0.5393544708}$$

cos(a,b) ≈ 0.964019

Calculate the neutrosophic Manhattan Distance between A and B using equation given in 6.3.

$$d(a,b) = |0.8 - 0.6| + \quad |0.1 - 0.2| + |0.1 - 0.2|$$

$$d(a,b) = |0.2| + \quad |0.1| + |0.1|$$

$$d(a,b) = 0.4$$

These measures are used in conjunction with Facenet[59] face recognition algorithm to improve the robustness and accuracy of face recognition systems over traditional methods. Traditional methods rely on handcrafted features or geometric features. However, facial expressions are highly complex with delicate patterns in facial muscles and configurations. Facenet a Deep learning-based approache in conjuction with Distance-based Similarity Measure USING NEUTROSOPHIC SETS can capture the intricate details and variations in facial expressions more effectively. The convolutional neural network based Facenet with neutrosophic Distance-based Similarity Measure trained on a dataset of facial images is capable of learning hierarchical representations related to facial expressions, such as wrinkles around the eyes.

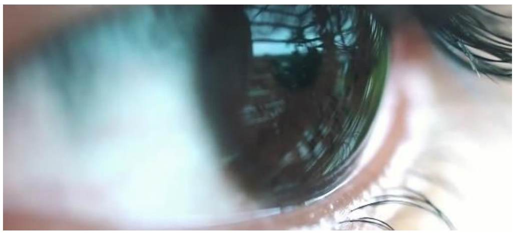
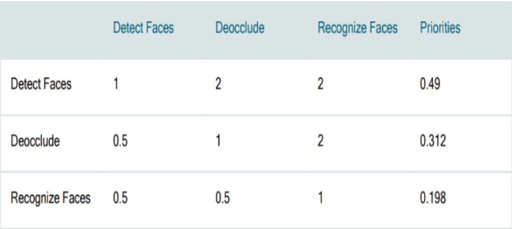


Figure. 15. Captured wrinkles around the eyes.

7. Neutrosophic AHP for prioritizing criteria

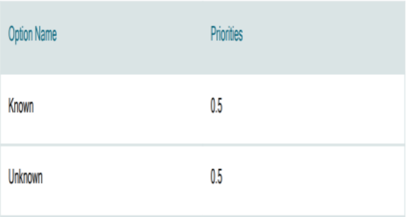
Analytic hierarchy process (AHP) is utilized to evaluate and prioritize various factors related to the problem statement of this research. Neutrosophic AHP[43] is used in prioritizing critereas for Video enhancement. Priority Trade-offs are set for criteria's like "Detect Faces, Deocclude and Recognize Faces. Priority values are selected based on equal, moderate and unequal importance.

Table. 4. Criteria Priorities



	Detect Faces	Deocclude	Recognize Faces	Priorities
Detect Faces	1	2	2	0.49
Deocclude	0.5	1	2	0.312
Recognize Faces	0.5	0.5	1	0.198

Table.5. Metrics



Option Name	Priorities
Known	0.5
Unknown	0.5

The consistency of the pairwise comparisons are made using consistency Ratio (CR). According To Thomas L.Saaty if the CR exceeds a predefined threshold (e.g., 0.1), the comparisons need to be revised to improve consistency. We obtained a Consistency Ratio calculated as 0.046. Pairwise Comparisons of Options for Criteria is done using the function below:

Multi-Criteria Utility Function = 0.49 * [Detect Faces] + 0.31 * [Deocclude] + 0.2 * [Recognize Faces]

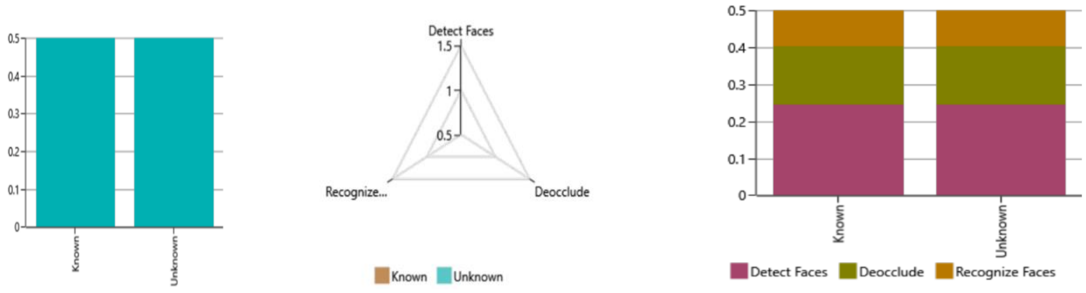


Figure. 16. a). Priorities b). Attributes c). Weighted Attributes

[Criterion : Recognize Faces]→[Option : Known]→[Value]

[Pair Comparison Against]→[Option : Unknown]

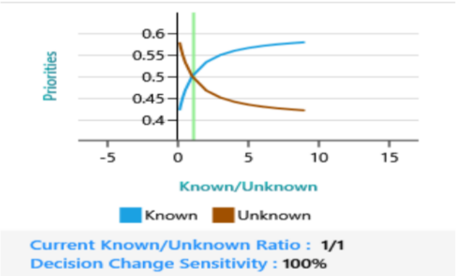


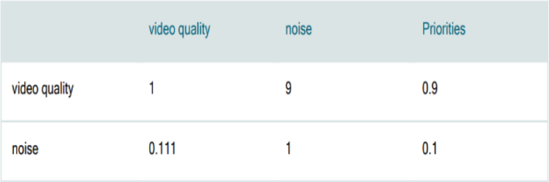
Figure. 17. Sensitive Variables

7.1. Indeterminacy Inherent Pixel Analysis Using Neutrosophic Analytic Hierarchy

Applying analytic hierarchy process (AHP) in neutrosophic sets to pixel analysis in our research aids in assessing and prioritizing various factors related to pixel-level characteristics and video frame processing tasks by taking into account uncertainties and indeterminacies in data. The objective is to optimize pixel analysis tasks such as Video enhancement, and denoise.

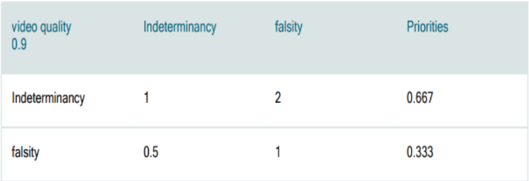
We begin by identifying two criteria's relevant to pixel analysis, such as video quality and noise. The criteria is organized in a hierarchical structure, with the goal at the top. Each criterion is paired with its corresponding parent criterion. Using the neutrosophic pairwise comparison results, the priority weights for each criterion is calculated. The consistency of neutrosophic pairwise comparisons are assessed using indeterminacy, and falsity metrics. We then aggregate the neutrosophic priority weights of criteria using to determine the overall ranking and prioritize pixel analysis tasks. Weight calculation using approximate Eigenvector and Priority Calculation using Weighted Sum: Weighted Sum = ∑(Weight * Attribute). Sensitivity analysis is performed to evaluate the pairwise comparisons considering the indeterminacy inherent in the decision-making process following results were obtained.

Table. 6. Priority Tradeoffs



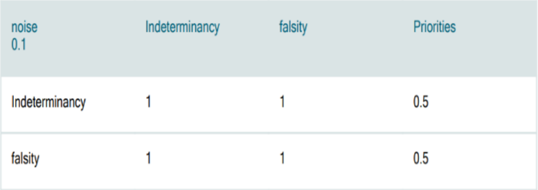
	video quality	noise	Priorities
video quality	1	9	0.9
noise	0.111	1	0.1

Table. 7. Comparisons for Criteria 'video quality'



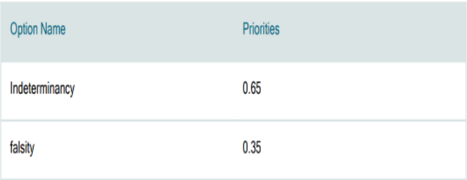
video quality 0.9	Indeterminancy	falsity	Priorities
Indeterminancy	1	2	0.667
falsity	0.5	1	0.333

Table. 8. Pairwise for Criteria 'video quality'



noise 0.1	Indeterminancy	falsity	Priorities
Indeterminancy	1	1	0.5
falsity	1	1	0.5

Table. 9. Metrics



Option Name	Priorities
Indeterminancy	0.65
falsity	0.35

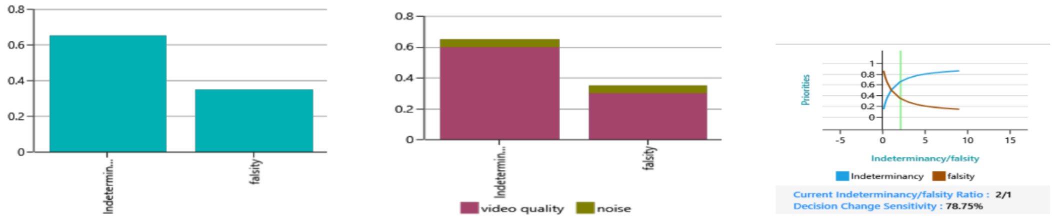


Figure.18. a). Priorities b). Weighted Attributes c). Sensitivity Variables

Multi-Criteria Utility Function = 0.9 * [video quality] + 0.1 * [noise] , Sensitive Variables (1): Criterion : video quality]→[Option : Indeterminancy]→[Value][Pair Comparison Against]→[Option : falsity]

8. Forensic Face Detection and Recognition Using Deep Learning

Face detection and recognition techniques are used in image and video analysis. They are widely used for making the crime investigation process easier. Existing face detection methods include, Face detection with Haar cascades which is extremely fast but prone to false-positives and in general less accurate than deep learning-based face detectors. Face detection with dlib (HOG and CNN) which is more accurate than Haar cascades but computationally more expensive. Dlib's CNN face detector is the most accurate of the bunch but cannot run in real-time without a GPU. Multi-task Cascaded Convolutional Networks (MTCNNs), which is very accurate deep learning-based face detector. MTCNNs is easily compatible with both Keras and TensorFlow.

8.1. Single Shot Detector

Single shot detector(SSD) is an object detection algorithm, it can detect multiple objects from a given image or a video stream in one shot. SSDs detect real time objects in video surveillance systems, selfdriving cars and Forensics. SSDs use a single pretrained network(CNN) to detect objects. The SSD functions upon ResNet base network. SSDs are better than R-CNN object detection algorithm which uses multiple networks and various stages to perform object detection. To this end, SSD is selected to detect objects in this work.

The CNN based ResNet model is chosen as the base network and SSD is made to work on top of it. The ResNets last layer which is a classification layer is truncated and convolutional feature layers are added. Convolutional predictors are used for detection. The model is compared with other object detection methods such as cf Overfeat[49] and YOLO[50] allows only single scale feature maps.

SSDs default boxes are similar to the anchor boxes used in Faster R-CNN [48]. The matching each ground truth box to the default box is made with the best jaccard overlap MultiBox [53]). Unlike MultiBox, in SSD default boxes are matched to any ground truth with jaccard overlap higher than a threshold (0.5) by simplifying the learning problem. The model loss is a weighted sum between localization loss, Smooth L1 [52] and confidence loss, Softmax.

The SSD training objective is derived from the MultiBox objective [53,54] but is extended to handle multiple object categories.

$$L(x,c,l,g) = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{loc}(x,l,g))$$

If N = 0, the loss is set to 0. The localization loss is a Smooth L1 loss [51] between the predicted box (l) and the ground truth box (g) parameters. Similar to Faster R-CNN [48], we regress to offsets for the center (cx, cy) of the default bounding box (d) and for its width (w) and height (h). The confidence loss over multiple classes confidences(c).

$$L_{loc}(x,l,g) = \sum_{i\in Pos}^{N} \sum_{m\in\{cx,cy,w,h\}} x_{ij}^{k} \mathrm{smooth}_{\mathrm{L1}}(l_i^m - \hat{g}_j^m)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h$$

$$\hat{g}_j^{w} = \log\left(\frac{g_j^w}{d_i^w}\right) \qquad \hat{g}_j^{h} = \log\left(\frac{g_j^h}{d_i^h}\right)$$

The confidence loss is the softmax loss over multiple classes confidences (c).

$$L_{conf}(x,c) = -\sum_{i\in Pos}^{N} x_{ij}^{p} log(\hat{c}_i^p) - \sum_{i\in Neg} log(\hat{c}_i^0) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$

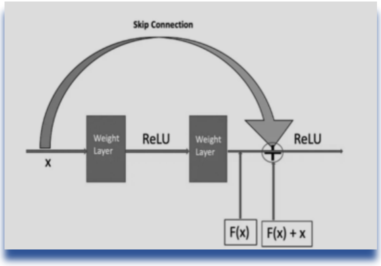
and the weight term α is set to 1 by cross validation.

The scale of the default boxes for each feature map is computed as: sk = smin + smax − smin m − 1 (k − 1), k ∈ [1, m] (49) where smin is 0.2 and smax is 0.9, meaning the lowest layer has a scale of 0.2 and the highest layer has a scale of 0.9, and all layers in between are regularly spaced.

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m-1}(k-1), \quad k \in [1,m]$$

8.2. Residual Neural Network

Deep networks often result in Gradients that vanishes as it is back-propagated to previous layers, repeated multiplication may make the Gradient infinitely small. ResNet[55] uses the concept of Residual blocks that include Skip connections.



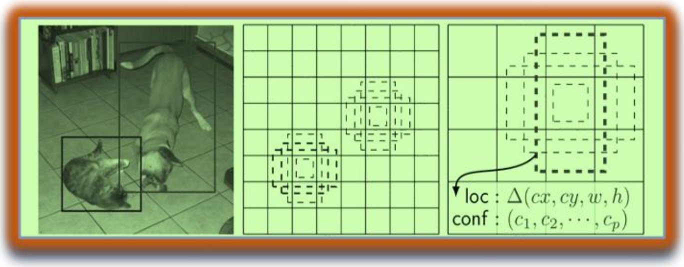

loc : $\Delta(cx, cy, w, h)$
conf : $(c_1, c_2, \cdots, c_p)$

Figure:19. a). Residual Block **b).** Feature Map

Figure. 19.b) Image with Ground-Truth boxes, 8 × 8 feature map and 4 × 4 feature map [51])

8.3. ResNet-34

ResNets[55] can easily be optimized and can gain increased accuracy by relative depth increment. The model is trained using FaceScrub and VGGFace2 datasets and tested with labeled faces in the wild (LFW) data set. The aligned faces are passed to the ResNet model and it represent faces 128 dimensional vector.

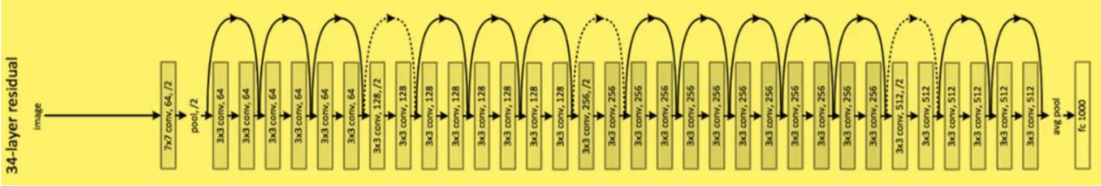


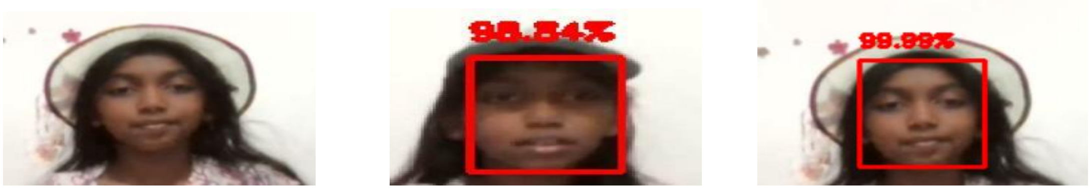
Figure. 20. ResNet Architecture

It consists of four residual blocks with three, four and six respectively. 64, 128, 256 and 512 Channels are used. Except for the first block, each block starts with a 3x3 kernel of stride of 2. Caffe deep learning framework is known for speed and its modularity[30]. Additionally, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes [31].

8.4. Implementation of Face detection Model

In this study for face detection, single shot detector framework with a ResNet base network is used which is trained using caffe deep learning framework. An image captured using webcam is displayed below. For detecting face from a captured image, a pre-trained face detection model built using ResNet base network, trained using caffe deep learning framework is used. The pre-trained face detection model, consists of ResNet base network definition and learned weights from caffemodel.

Initially the captured image is passed through the trained Res-Net base network for making detections and predictions. A loop construct is used to loop over the detections and to draw boxes around the detected faces. The confidence probability associated with the predictions are extracted. The minimum confidence threshold is selected to be 0.5. All the weak detections are filtered to ensure the `confidence` is greater than the minimum confidence threshold 0.5. If confidence is greater than minimum confidence threshold then the (x, y)-coordinates of the bounding box for the object is computed. The bounding box of the face along with the associated probability with (x, y) coordinates are drawn. The model being trained on different captured images has detected face accurately at 99.88% similarity.



(a) (b) (c)

Figure 21. Face detection using single shot detector framework with ResNet base network produces results as listed: (a) Captured image; (b) Face detected with 98.84% accuracy; (c) Face detected with 99.88% accuracy

8.5. Face Recognition Using Deep Learning

A facial recognition system[1] is a technology potentially capable of matching a human face from a digital image or a video frame against a database of faces. This can be done through an eyewitness or from digitally stored pictures. Facial recognition software uses specific points on an image, compares those points to the same points of images in a database and finds similarity percentage to recognize faces.

Face_recognition library is used in this paper for implementing face recognition module built using dlib. Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is used in a wide range of applications including robotics, embedded devices, and large high performance computing environments [32]. To run Face_recognition module efficiently AMD Ryzen RTX 3050 GPU is used. The module operates on a facedataset which consists of both known and unknown face images. First an input image is loaded into the model, and the model generates 128 face encodings for the loaded image. Faces which are similar will have same encodings and for faces which are different the encodings will differ. The length of the encodings is verified. If the length of encodings is greater than one then the model has detected more than one face from the input image. If the length of the encoding is equal to zero then no faces are detected from the input video and hence the file is ignored. If the length of encodings is neither 0 nor greater than one then appends the base image name to known_names array and appends the image encodings to know_face_encodings array. First an unknown image is loaded to test the module for face recognition task.

8.5.1. Dlib - Face Recognition Library

Dlibs face recognition pipeline consists of 4 common stages: detect, align, represent and verify. Dlib is mainly inspired from a ResNet-34 model[55]. Davis E. King in his work modified the regular ResNet structure and dropped some layers and re-build a neural network consisting of 29 convolution layers. Accepts 150x150x3 size input and represent 128 dimensional vectors. Dlib is capable of finding 68 facial landmark points on the face including eyes, eye-brows, mouth, lips, jaw, chin, and nose. Face detection does not have to be applied for rectangle areas as Haar cascade does.

Using Dlib we are able, to do out-of-the-box face recognition as well. Sixty eight landmarks of dlib is shown below.

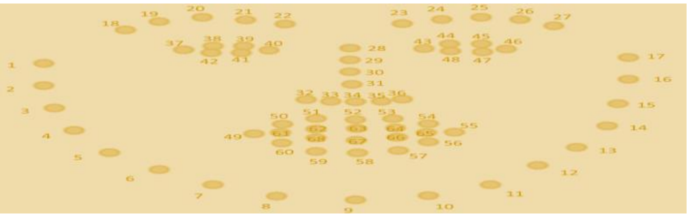


Figure. 22. Sixty Eight different landmarks of a face

8.5.2. Face Recognition implementation

The loaded image is preprocess to resize the image if its size is greater than 1600. The image is resized using thumbnail() resizing filter which takes a resampling filter called lanczos kernel as its parameter. It is the normalized sinc function sinc(x), windowed (multiplied) by the Lanczos window, or sin c window, which is the central lobe of a horizontally stretched sin c function sin c(x/a) for −a ≤ x ≤ a. After resizing, the image is converted into array of values. Next for unknown input image generate encodings. Calculate distance between known face encodings and unknown face encodings. The tolerance is set to 0.5. Then all images whose distances are less than or equal to tolerance are extracted. The details of image name and distance from all extracted images are retrieved. The image with less distance is considered to be more similar to the search image. The deep learning module has successfully detected and recognized faces. Face detection and recognition of Radullan Sahiron and Ahmad abousamra is successful by the face_recognition module as shown in figure 12.

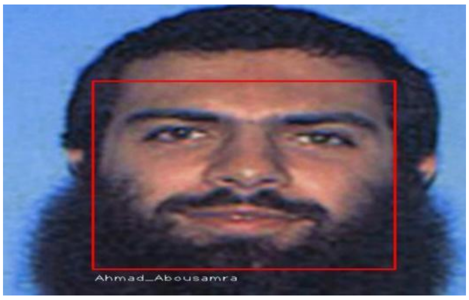


Figure 23. Deep learning Face_recognition module: Detected and recognised image of Ahmad Abousamra with 98.84% accuracy.

Dlib finds representations in dlib.vector type, to find the Euclidean distance using tuned threshold between images we can convert it to numpy in order to find the similar faces with threshold value 0.5 which results in 99.88% confidence score on LFW data set. On the other hand, human beings hardly have 97.53% score on same dataset. Dlibs face recognition model competes with the other state-of-the-art face recognition models and human beings as well.

9. Advantages of Amd Ryzen Rtx 3050 Gpu in Face Detection and Recognition Task

The AMD Ryzen RTX 3050 GPU offers several advantages for face detection and recognition tasks, especially when paired with implementations like SSD (Single Shot Multibox Detector), ResNet (Residual Neural Network), and dlib. The proposed model with NSS, SSD, ResNet, and dlib implementations with GPU acclerations is capable of performing complex convolutions and extract facial features, by passing through deep ResNets, resulting in faster inference times. NVIDIA's optimized Tensor Cores, present in the RTX 3050 GPU, accelerates deep learning workloads which makes suitable for face detection and recognition. The RTX 3050 GPU offers improved performance over previous generations, with higher clock speeds, more CUDA cores, and enhanced memory bandwidth. This translates to faster inference speeds and better overall performance for face detection and recognition tasks. The deep learning frameworks, TensorFlow and PyTorch, have CUDA support. CUDA, is developed by NVIDIA for GPU-accelerated computing. By implementing the work in tensorflow we are able to leverage the GPU's processing power which results in enhanced performance of SSD, ResNet, and dlib implementations on the RTX 3050 GPU. RTX 3050 GPU with its VRAM (Video Random Access Memory) feature with its adequate memory capacity supports us to efficiently run face detection and recognition model with smaller datasets.The RTX 3050 GPU's scalability depends on the specific application requirements and dataset sizes. The GPU handles real-time inference for face detection and recognition tasks efficiently working with smaller to moderate-sized datasets. NVIDIAs RTX 3050 GPU combined with AMDs RDNA architecture, results in better energy efficiency compared to previous generations. The GPU consumes less energy which makes the model suitable to be deployed in systems with power consumption concern. The AMD Ryzen RTX 3050 GPU offers a promising solution for face detection and recognition tasks, particularly when used with SSD, ResNet, and dlib implementations. Its GPU acceleration, optimized tensor cores, increased performance, CUDA support, memory capacity, scalability, and energy efficiency make it well-applicable for small to medium datasets and potentially for forensics real-time applications.

10. Key Contributions

A Novel Filter is using Sliding Window Convolution is proposed for Image Denoising. Sliding window is also called as Rolling Window. Using Sliding window concept we first highlight the pixels to be operated by the kernel. Convolution is faster with sliding window since there is no loop. The creation of the rolling window does not take much time because no data is copied, it is just a view on the original matrix. An image affected with impulse noise is taken as input. For each sliding window first the pixels are sorted and the Mean(r) pixel is calculated. When r=1 the filter functions similar to median filter. If r=3 then novel filter function is invoked. The output is compared with median filter. The obtained result as in figure 11 has 99% similarity to that of median filter. For image and video frame sharpening a TripleMask Spatial Linear filter is proposed. TripleMask Spatial Linear filter is created by cascading a series of filters. Identity, Shift and Fraction based approach is used in Mask processing. For Image smoothing and to preserve the edges Bilateral filter is used [2]. The results obtained are shown in Figure.4. The performance of convolution operation is compared as in table.3 in terms of CPU time with distinct convolution, shift rotational convolution, sliding window convolution and scipy convolution. We address the Video quality issues issues involving shaky or unstable camera movements which leads to blurry footage. In our work we use a Hybrid model for video deblurring using Deep Video Deblurring for Hand-held Cameras[56], DEblurGAN[57] and Spatio-Temporal Transformer Networks for Video Deblurring[58] and proposed TripleMask spatial linear filter. Through the combined approach and leveraging the power of deep learning our model is capable to surpass traditional methods by performing a data-driven mapping from blurry to sharp frames, to learn complex motion patterns, to generate high-quality deblurred results, to obtain superior deblurring performance, mainly for dynamic scenes. Results produced by the combined approach using set of filters including TripleMask spatial linear filter, Median Filter and Bilateral filters are shown below. Sharpening the video frames at an amount of 0.50, the model enables to see clearly the objects in the frame. Deblurring the frame to see very minute details at an amount of 0.78. By applying median filter by setting mean filter threshold to 50, edge threshold 50, motion threshold 50 we get a more clear visual appearance. We obtain temporal softening with radius 5, luma threshold 4 and croma threshold 8 which causes a smoothing effect and removes fluctuations in pixel values over time. This helps reduce noise and improve the overall quality of the video. By turning right or left and changing the colors of the object it enables to visualize in different directions to further recognize the object. It smooths the video frame by preserving edges by considering both the spatial distance and the intensity difference between neighboring pixels. Median Filter: threshold:50, Edge threshold:50, Motion threshold:50. Temporal Soften: radius 10, luma threshold 10 and croma threshold 20 .

Figure. **24**. a). Sharpen scale:1.00 b). Blur scale:1.00 c). Median Filter

Figure. **25**. a). Temporal Soften b). TripleMask Spatial linear Soften

10.1. Video Frame processing and analysis using Histogram

A histogram is a graphical representation that displays the distribution of data. It consists of a series of vertical bars, where each bar represents the frequency or count of data values falling within a particular range or interval. Histograms are commonly used in statistics, data analysis, and image processing to visualize the distribution of numerical data. In our work we use histogram for video processing to represent the distribution of pixel intensities within a frame. Each pixel in a frame has an intensity value, which can range from 0 (black) to 255 (white) in an 8-bit grayscale image. The histogram displays the frequency of occurrence of each intensity value within the image.

Histogram representations are used to perform analysis and understand the image in terms of its colour, contrast, and brightness. Each pixel in an image has red, green, and blue colour components, and the red channel histogram shows the frequency of different red intensity levels present in the image. Green channel histogram shows the frequency of different green intensity levels present in the image. Blue channel histogram shows the frequency of different blue intensity levels present in the image. Histogram representations are used in this research work to assess color balance, identify color casts, adjust pixel intensity levels to obtaing correct color balance and to achieve more accurate color reproduction.

Figure. **26**. Original Frame

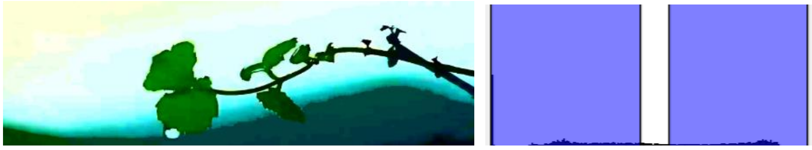


Figure. **27**. Low:120, Gamma: 1.76 High: 144 and blue histogram

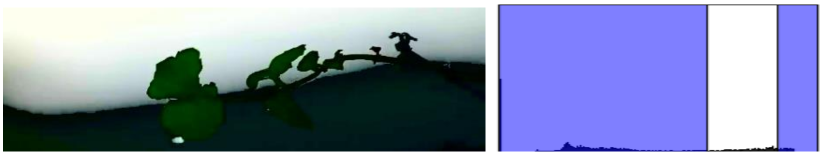


Figure. **28**. Low:166, Gamma: 3.16 High: 223 and Green histogram

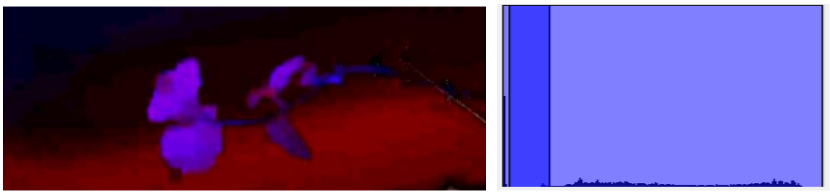


Figure. **29.** Low:38, Gamma: 1.16 High: 6 and Red histogram

10.2. Traditional entropy measures

Traditional entropy measures come from information theory and statistical mechanics and are used to quantify the uncertainty or disorder in a system. The most common traditional entropy measures include:

10.2.1 Shannon Entropy (Information Entropy): Introduced by Claude Shannon in his seminal work on information theory, Shannon entropy quantifies the uncertainty associated with a random variable. It is defined as the average amount of information produced by a random variable. Mathematically, for a discrete random variable X with probability mass function p(x), Shannon entropy H(X) is given by: $H(X) = -\Sigma\ p(x) * log2(p(x))$

10.2.2 Boltzmann Entropy: In statistical mechanics, Boltzmann entropy is a measure of the microscopic disorder or randomness of a system. It's defined as: $S = kB * ln(W)$

Where S is the entropy, kB is the Boltzmann constant, and W is the number of microstates corresponding to a given macrostate.

10.2.3. Gibbs Entropy: In thermodynamics, Gibbs entropy is a measure of the disorder in a system at equilibrium.

It's defined as: $S = -k * \Sigma\ pi * \ln(pi)$, Where S is the entropy, k is the Boltzmann constant, and pi is the probability of the system being in the i-th microstate.

10.2.4 Renyi Entropy: It is a generalization of Shannon entropy and Boltzmann entropy, which introduces a parameter α. When α approaches 1, Rényi entropy converges to Shannon entropy.

It's defined as: $H\alpha(X) = 1/(1-\alpha) * \log2(\Sigma\ p(x)^\wedge\alpha)$

Where p(x) is the probability mass function of the random variable X.

10.3. *Selection Of Neutrosophic Entropy in Feature Selection Instead of Traditional Entropy Measures*:

Neutrosophic entropy considers truth-membership, indeterminacy-membership, and falsehood-membership degrees, allowing it to capture uncertainty in feature selection more comprehensively compared to traditional entropy measures. For performing Image/Video frame analysis and for making feature selection in this work Neutrosophic Entropy is used. For forensic face feature detection Neutrosophic Entropy values are used. The input image is loaded in grayscale. Image patches are created. For each image patch Histogram values are created using calcHist method of opencv.

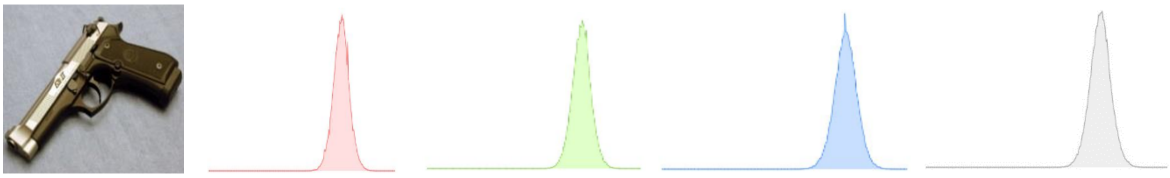


Figure. 30. Image, Histogram: Red, Green, Blue and Luminosity

The calculated histogram is normalized using equation, x=x/x.sum(). Then entropy values are calculated for each patch based on its histogram created using the following calculation.entropy = (hist * np.log2(hist + np.finfo(float).eps)). The patches with the highest entropy and lowest entropy is selected, visualizes or processed to detect face features.

Histogram analysis is often used for automatic or manual determination of threshold values for image segmentation. In this paper thresholding is applied to identify indeterminate regions. An indeterminacy_filter is defined to work the input image with threshold =100. Then the binary image is inverted to get indeterminate regions. Neutrosophic_entropy is calculated even for checking individual pixel_memberships >0, initialy entropy is set to be 0.0 and a small constant epsilon, to avoid taking the logarithm of zero is used, where epsilon = 1e-10. for membership in pixel_memberships, if membership > 0 then entropy is calculated as entropy -= membership * np.log(membership + epsilon). It is observed for those pixels having intensity values with equal probability, the entropy is maximum, and for pixels having low intensity the entropy values are low indicating nonuniform intensity distribution. Pixel at (10,60) has intensity: 2 resulting with low entropy ((10, 60), 0.8812909). From the calculated entropy values, it is observed, when pixels having low intensity its entropy values are also low. Such pixels with low entropy values has nonuniform intensity distribution in the input image. Such finding helps in finding crucial image regions. By using distance-based similarity measures using neutrosophic sets in face recognition we are able to find uncertainty and imprecision inherent in facial feature representations. Region-of-Interest enhancement is performed using image entropy. The produced result shows Increased color contrast for the region of interest: x, y, width, height = 100, 150, 200, 200, Extracted region of interest using arrayslicing, Color contrast increased using convertScaleAbs for the region of interest coordinates x, y, width, height = 100, 50, 200, 200 in figure 15. Face detection using single shot detector framework with ResNet base network has produced results as shown in figure. 22, Captured image, Face detected with 98.84% accuracy and Face detected with 99.88% accuracy. Deep learning Face_recognition module result is shown in figure .23 , Detected and recognised image of Ahmad Abousamra with 98.84% accuracy.

11. Future Research Directions

Neutrosophic sets can be employed to delineate image regions or objects based on their degrees of truth, indeterminacy, and falsity. As a challenging task neutrosophic set can be used in uncertain boundaries of images in the process of image segmentation. Neutrosophic entropy value can provide information about how much information is contained in the neutrosophic set representation of the image. Neutrosophic set entropy can be used in image segmentation and clustering oriented problems. Researchers can apply neutrosophic set for detecting faces from a video. Neutrosophic methods can be employed in risk assessment and management, especially when dealing with complex and uncertain risks in fields like insurance and project management. Neutrosophic logic can be integrated into AI and ML models to handle uncertain or contradictory data. Neutrosophic approaches can be used to assess and manage environmental data that is often uncertain or incomplete. This can be particularly used in climate modeling and ecological studies. In our work we have only used pre-recorded videos for processing and to detect faces and recognize them from a set of manually collected videos using AMD Ryzen RTX 3050 GPU with limited memory capacity. But in future the proposed framework can be extended to work with real time surveillance systems to detect faces and recognize for forensic investigations. Also high end GPUs can be used instead of AMD Ryzen RTX 3050 GPU to make the system compatible to work with dynamic large scale datasets.

12. Conclusions

Video quality issues such as blurry, speckled, pixelated and low-resolution videos captured at low light are a real challenge in forensic analysis. Such issues are addressed in this research using a set of algorithms and techniques. For denoising images, a novel image filter using sliding window convolution is proposed and used. For image sharpening a TripleMask Spatial Linear filter is proposed and applied. TripleMask Spatial Linear filter is created by cascading a series of filters. Identity, Shift and Fraction based approach is used in Mask processing. For Image smoothing and to preserve the edges Bilateral filter is used [2]. The performance of convolution operation is compared with distinct convolution, shift rotational convolution and scipy convolution. To handle uncertainty, imprecision, and ambiguity in real-world image data in a precise manner neutrosophic science is used in image analysis. By the generated Neutrosophic set of the given input image ambiguous regions in the image are detected. Feature selection is made by calculating the entropy of different image regions. From the generated Neutrosophic set entropy the degree of uncertainty, within the input image is quantified. The intensity distributions are measured using entropy values. In Feature selection regions with highest and lowest entropy values containing face images are selected, visualized and processed to further aid in forensic analysis in detecting the culprits. Face detection is performed using single shot detector framework with a ResNet base network, trained using caffe deep learning framework. Face recognition process is performed using dlibs [9] state-of-the-art face recognition model built with deep learning [10]. The model was successfully implemented using AMD Ryzen RTX 3050 GPU. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark.

Funding: “This research received no external funding”

References

1. K. Sita Manikyam, Forensic Science and its Limitations in Rape and Murder Cases in India, March 2023, *Journal of Forensic Science and Medicine* 9(1):91-97
2. Sylvain Paris; Pierre Kornprobst; Jack Tumblin; Frédo Durand, Bilateral Filtering: Theory and Applications, now, 2009.
3. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," San Diego, CA, USA, 2005, pp. 886-893 *vol. 1*, doi: 10.1109/CVPR.2005.177.
4. Kazemi, Vahid ,One Millisecond Face Alignment with an Ensemble of Regression Trees, 2014/06/01
5. Hassan, Rondik, Deep Learning Convolutional Neural Network for Face Recognition: A Review, 2021/01/27
6. Triantafyllidou, Danai & Tefas, Anastasios, 3560-3565. 10.1109/ICPR.2016.7900186.
7. FaceNet, https://doi.org/10.48550/arXiv.1503.03832
8. Brandon amos,Bartosz Ludwiczuk, Mahadev Satyanarayanan, OpenFace: A general-purpose face recognition library with mobile applications, 1 Jan 2016
9. Davis E. King. Dlib-ml: A Machine Learning Toolkit.
10. Adam Geitgey, Face Recognition, Copyright 2017, Adam Geitgey Revision 2e2dccea.
11. Geitgey, *A.,Deep Learning: Face Recognition, BOOK,* 2018.

12. Mark M. Pollitt, Handbook of Digital and Multimedia Forensic Evidence, 2008, ISBN : 978-1-58829-782-2

13. Forensic science services and the criminal justice system, published Friday, 12 November, 2021

14. The Guardian, Forensic Science Failures putting jusice at risk, February 2020

15. Graeme Horsman, Digital evidence and the crime scene, Science & Justice, Volume 61, Issue 6, 2021, Pages 761-770, ISSN 1355-0306.

16. Video Evidence – A Law Enforcement Guide to Resources and Best Practices , March 2014.

17. Abdul Rehman Javed, Zunera Jalil, Wisha Zehra, Thippa Reddy Gadekallu, Doug Young Suh, Md. Jalil Piran, A survey on digital video forensics, *Volume 106*, 2021, 104456, ISSN 0952-1976.

18. Sowmya K.N., H.R. Chennamma, A Survey On Video Forgery Detection, arXiv:1503.00843

19. J. Xiao et al.: Video-Based Evidence Analysis and Extraction in Digital Forensic Investigation

20. Kim, J., Lee, D. & Park, N. Multimedia Tools Appl 79, 23461–23481 (2020). https://doi.org/10.1007/s11042-020-09016-z

21. Kimberley A. Wade, Sarah L. Green, Robert A. Nash, Can fabricated evidence induce false eyewitness testimony? Applied Cognitive Psychology, 24(7), 899-908 - August

22. 2009K. A. Wade, S. L. Green, & R. A. Nash, from the archive: 'can fabricated evidence induce false eyewitness testimony?' by (2010), 24, 899–908 with commentary

23. JNina J. Westera - Griffith University, Martine B. Powell - Deakin University, Prosecutors' perceptions of how to improve the quality of evidence in domestic violence cases Policing & Society, 27(2), 157-172 - June 2015, https://doi.org/10.1080/10439463.2015.1039002

24. Nina J Westera - Griffith University, Lost in the detail: Prosecutors' perceptions of the utility of video recorded police interviews as rape complainant evidence, Journal of Criminology, 50(2), 252-268 - December 2015, https://doi.org/10.1177/0004865815620705

25. Annamaria Motrescu-Mayes, Susan Aasman, Amateur Media and Participatory Cultures, Film, Video, and Digital Media, January 2019, https://doi.org/10.4324/9781315396149

26. https://en.wikipedia.org/wiki/Kernel_%28image_processing%29

27. https://web.pdx.edu/~jduh/courses/Archive/geog481w07/Students/Ludwig_ImageConvolution.pdf

28. https://www.cs.cornell.edu/courses/cs1114/2013sp/sections/S06_convolution.pdf

29. https://docs.scipy.org/doc/scipy/tutorial/ndimage.html

30. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, Trevor Darrell, Caffe: Convolutional Architecture For Fast Feature Embedding, https://doi.org/10.48550/arXiv.1408.5093

31. Wei liu, Dragomir anguelov, Dumitru erhan, Christian szegedy, Scott reed, Cheng-yang fu, A, SSD: Single Shot MultiBox Detector, https://docs.scipy.org/doc/scipy/tutorial/ndimage.html

32. Davis E. King. Dlib-ml: A Machine Learning Toolkit, Journal of Machine Learning Research, 2009, *volume 10*, pages 1755-1758.

33. Haitham A. El-Ghareeb: Novel Open Source Python Neutrosophic Package, *Neutrosophic Sets and Systems, vol. 25*, 2019, pp. 136-160, http://fs.unm.edu/NSS/NovelOpenSourcePython.pdf.

34. M. R. Faraji and Xiaojun Qi, "An effective neutrosophic set-based preprocessing method for face recognition," 2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), San Jose, CA, USA, 2013, pp. 1-4, doi: 10.1109/ICMEW.2013.6618251.

35. A neutrosophic approach to image segmentation based on watershed method, *Signal Processing, Volume 90*, Issue 5, 2010, Pages 1510-1517, ISSN 0165- 684.

36. Mohammad Reza Faraji and Xiaojun Qi, An Effective Neutrosophic Set-Based Preprocessing Method For Face Recognition, Http://fs.unm.edu/NeutrosophicPreprocessing.pdf

37. A. Salama, Florentin Smarandache and Mohamed Eisa, Introduction to Image Processing via Neutrosophic Techniques, Neutrosophic Sets and Systems, Vol. 5, 2014.

38. Bolón-Canedo, V., Remeseiro, B. Feature selection in image analysis: a survey. Artif Intell Rev 53, 2905–2931 (2020). https://doi.org/10.1007/s10462-019-09750-3

39. Broumi, S., S. krishna Prabha, & Vakkas Uluçay. (2023). *Neutrosophic Systems With Applications*, *11*, 1–10. https://doi.org/10.61356/j.nswa.2023.83

40. Norzieha Mustapha1 , Suriana Alias2*, Roliza Md Yasin3 , Ilyani Abdullah4 and Said Broumi, cardiovascular diseases risk analysis using distance-based similarity measure of Neutrosophic Sets

41. Jdid, M., & Smarandache, F. (2023). Optimal agricultural land use, using Neutrosophic Linear Programming Method. *Neutrosophic Systems With Applications*, *10*, 53–59. https://doi.org/10.61356/j.nswa.2023.76

42. Mohamed, M., Karam M. Sallam, & Ali Wagdy Mohamed. (2023). Innovations of Industry 5.0 Technologies in smart supply chain partners. *Neutrosophic Systems With Applications*, *10*, 1–11. https://doi.org/10.61356/j.nswa.2023.74

43. Ahmed A. El-Douh, SongFeng Lu, Ahmed Abdelhafeez, Ahmed M. Ali, & Alber S. Aziz. (2023). Heart disease prediction under machine learning and association rules under Neutrosophic Environment. *Neutrosophic Systems with Applications*, *10*, 35–52. https://doi.org/10.61356/j.nswa.2023.75

44. Vakkas, U., Kılı¸c, Sahin, M., Deniz, H. Distance-based similarity measure for refined neutrosophic sets and its application in Medical Diagnosis, Matematika, 2019, 35(1), 83–96.

45. Guleria, A., Srivastava, S., Bajaj. R.,K. Application in Decision-making Models, Neutrosophic Sets and Systems, 2019, 29, 101–120

46. Broumi, S.; Smarandache, F. Several Similarity Measures of Neutrosophic Sets, Neutrosophic Sets And Systems, 2013, 1(1), 54–62.

47. Majumdar, P.; Samanta, S. K. On Similarity and Entropy of Neutrosophic Sets, Journal of Intelligent and Fuzzy Systems, 2013, I, 1–13.

48. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS. (2015)

49. Sermanet, P. Eigen, D. Zhang, X. Mathieu, M. Fergus, R. LeCun, Y. Overfeat, Integrated recognition, localization and detection using convolutional networks. In: ICLR. (2014)

50. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. (2016)

51. Wei Liu, Dragomir Anguelov , Dumitru Erhan , Christian Szegedy , Scott Reed , Cheng-Yang Fu, Alexander C. Berg, SSD: Single Shot MultiBox Detector

52. Girshick, R.: Fast R-CNN. In: ICCV. (2015)
53. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: CVPR. (2014)
54. Szegedy, C., Reed, S., Erhan, D., Anguelov, D.: Scalable, high-quality object detection. arXiv preprint arXiv:1412.1441 v3 (2015)
55. [1512.03385] Deep Residual Learning for Image Recognition - arXiv.org, DOI: https://doi.org/10.48550/arXiv.1512.03385
56. Seungjun Nah Tae Hyun Kim Kyoung Mu Lee, Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring Department of ECE, ASRI, Seoul National University, CVPR 2017
57. Orest Kupyn, DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks, 2018
58. Jinshan Pan et al., Deep Discriminative Spatial and Temporal Network for Efficient Video Deblurring Nanjing University of Science and Technology.
59. Schroff, Florian, FaceNet: Unified Embedding for Face Recognition and Clustering, DOI: 10.1109, 2015, 7298682

Received: Feb 2, 2024. Accepted: April 26, 2024