REPUBLIC OF TURKEY

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

IMAGE QUALITY ENHANCEMENT AND RAIN REMOVAL ON IMAGES TAKEN UNDER DIFFERENT RAIN CONDITIONS

Ahmed Fraidoon ABDULKAREM

Master's Thesis

Supervisor

Prof. Dr. Ayça Kurnaz Türkben

Istanbul, 2022

IMAGE QUALITY ENHANCEMENT AND RAIN REMOVAL ON IMAGES TAKEN UNDER DIFFERENT RAIN CONDITIONS

Ahmed Fraidoon ABDULKAREM

Information Technologies

Master's Thesis

ALTINBAŞ UNIVERSITY

2022

The thesis titled “COMPUTER NETWORK TRAFIC CLASSICATION BASED AL TECHNIQUES prepared by AHMED FRAIDOON ABDULKAREM and submitted on 20/05/2022 **accepted unanimously** for the degree of Master of Science in Information Technologies.

____

Asst. Prof. Dr.Ayca Kurnaz Turkben

Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr.Ayca Kurnaz Turkben	Faculty of,Engineeing and Natural Altinbas University	________________
Asst. Prof. Dr. Abdullah Abdu IBRAHIM	Faculty of Engineeing and Natural Altinbas University	________________
Asst. Prof. Dr. Serdar KARGIN	Faculty of Engineeing and Natural University	________________

I hereby declare that this thesis/dissertation meets all format and submission requirements of a master’s thesis.

Submission date of the thesis to Institute of Graduate Studies: ___/___/___

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Ahmed ABDULKAREM

Signature

DEDICATION

With honor, I salute the memory of my late parents.

PREFACE

God has given me the patience and strength, as well as praise and acknowledgment, to do my duty.

In addition to Prof. Dr. Ayça Kurnaz Türkben, I'd like to thank him for his steadfast support and guidance throughout the project, as well as his inspiration, patience, and extensive knowledge. It's his constant encouragement and support that have helped me become more self-reliant while working on my thesis. I hope he succeeds in every attempt. The members of my thesis committee were very helpful and gave me very good advice. Please accept my heartfelt thanks for their help and advice.

Everyone at Kirkuk Hospital, as well as others who assisted me in putting my study into action, owes me an enormous amount of appreciation. I could not have finished this research without their help, which was immeasurable.

I owe a debt of gratitude to all of my loved ones and friends who have always encouraged and supported me in my quest for a college degree.

Last but not least, I'd like to say thank you to all of my Master's classmates who helped me with my spiritual well-being as I researched and wrote my thesis.

ABSTRACT

IMAGE QUALITY ENHANCEMENT AND ANALYSIS TAKEN BY MULTIPLE CAMERAS UNDER RAIN CONDITION

Abdulkarem , Ahmed

MSc, Information Technologies , Altınbaş University,

Supervisor: Prof. Dr. Ayça Kurnaz Türkben

Date: May / 2022

Pages: 61

The visual quality of images recorded on rainy days, even with high-resolution cameras, is worse, which hampers analytical tasks such as object recognition and categorization. Because of this, picture de-raining has been a hot topic in recent years. An artificial neural network is proposed for the de-raining, contrast enhancement, and feature analysis of a single picture in this study. Deterioration of a picture is judged in comparison to the perfection of a picture. We introduce a deep convolutional neural network (CNN)-based BCET for improving contrast and deep network architecture for removing rain for bettering the image quality. GLCM and DWT feature extractions are used for quality assessments as well. A deep residual network (Res-Net) inspired us to develop a deep detail network that immediately reduces the mapping distance from source to destination, making the learning process easier. The model is trained with a high resolution that decreases background interference and concentrates the model on the structure of rain in photos using previous knowledge of the picture domain to enhance the de-rained output. Contrast enhancement and rain removal have been employed to study image features. The significant power swings in visuals caused by rain diminish the quality and usefulness of outdoor vision systems. Both on images that are made up and images that are real, the presented strategy is better than current best practices in terms of both quality and number.

Keywords: Image Processing, Feature Extraction, Deep Learning, Rain Removal, Contrast Enhancement.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

CNNs	:	Convolutional Neural Networks
PSNR	:	Peak Signal-To-Noise Ratio
PRNU	:	Photo Responses Non-Uniformity
FPR	:	False Positive Rate
SVM	:	Support Vector Machines
GLCM	:	Gray Level Co-Occurrence Matrix
DWT	:	Discrete Wavelet Transform:
MSR	:	Multi-Scale Retinex
CBET	:	Contrast Balance Enhancement Technique
CLAHE	:	Contrast Limited Adaptive Histogram Equalization
Res-Net	:	Deep Residual Network
DDN	:	Deep Detail Network
SSIM	:	Structure Similarity Index
YOLOV3	:	You Only Look Ones
RNNs	:	Recurrent Neural Networks

1. INTRODUCTION

Rain, the most common kind of inclement weather, may degrade the clarity of images and harm outdoor vision equipment. When it's raining, rain streaks provide a hazy appearance in images, but they may also cause light to scatter. Getting rid of rain streaks is important in a lot of different situations, like when you want to improve a picture or track an object.

Image processing techniques may be harmed by adverse weather conditions. The outputs of these algorithms are generated using feature information from the images taken by different sources. As a consequence, it's vital to create algorithms that can resist weather changes. There are currently two types of rain removal methodologies: for videos and still photos. Because images under rain have redundant temporal information, rain streaks may be more easily detected and removed [1] – [4]. For example, the researcher in [1] initially suggest a correlation-based rain streak identification technique. After determining the position of rain streaks, the approach removes streaks by averaging pixel values from surrounding frames. A frequency-space description of rain's visual effect has been developed by Barnum et al. A distribution of streak direction is used to determine rain in [3]. In [4] uses phase consistency to locate and delete rain streaks by reducing registering error across frames. Many of these techniques are effective, but video's temporal element makes them much more so. Instead, we concentrate on eliminating rain from a single photograph in this work.

The authors of [5] suggested a motion segmentation-based method for dynamic scene segmentation. They identified rain using photometric and color constraints, then applied rain removal masks on pixels using dynamic characteristics and motion interference cues, recovering rain pixels using both spatial and temporal information.

The hybrid feature set [6] was utilized to reduce rain while boosting non-rain components. Image deconstruction and self-learning approaches were investigated [7] for image quality enhancement and rain removal.

The method in [7] initially performs context-constrained image identification. Several sorts of dictionaries will swap atoms that fit rain patterns. To remove rain as a high-frequency component, the technique [7] employs classifiers on acquired language sets to automatically predict common rain patterns. Using MCA, Fu, Yu-Hsiang, et al. [8], all of the foregoing approaches, however, are inadequate and need specialized high-level knowledge. Everyone knows that a camera's principal function is to take photos, yet the concept of picturing has developed since the early black and white cameras. The objective is to take a 2D photo after removing the darkness from the item. A 2D image has coordinates specified by spatial domain locations [9].

A digital picture has a limited number of components, each with a unique amplitude and position, known as image elements, or image pixels. The picture pixel represents the digital image components. Therefore, digital image processing has a broad range of applications [10].

Picture resolution refers to the number of pixels in a digital image. With any lens setting, the smaller the pixel size, the better the resolution and the clearer the picture. Images with reduced pixel sizes may have more pixels. The number of pixels corresponds to the image's information. Pixel resolution is expressed as 7680 by 6876, where the first number indicates the picture width or pixel columns and the second number represents the image height or pixel rows. Other standards include describing pixels per dimension or area unit (e.g., pixels per inch), etc. A picture of 1024 x 2048 pixels contains 2097152 pixels [10].

Pre-processing images (quality improvement and analysis), increasing the core vision algorithms' robustness, and supplementing the sensor module with multi-modal sensors are three strategies to cope with rainy weather in automated surveillance of the image. We will examine the effects of enhancement and analysis raw images data to lessen and analyze the dynamic effects of rain and snowfall in this work. They say their strategies may help standard vision approaches like fragmentation, identification, and monitoring become more robust. We will investigate this claim quantitatively, bringing new insights to the academic community.

Currently, rain removal techniques are assessed using recordings or still photographs given by the authors. Rain streaks are added to rain-free pictures to create synthetic datasets [11]. Indoor photos containing synthetic rain are often seen as part of rain removal algorithm training [12] and testing [13].

1.1 PROBLEM STATEMENT

It's possible that rain will reduce picture quality and impair outdoor machinery's ability to see. Rain streaks generate light dispersion in photographs, resulting in haziness, when it rains. Many beneficial applications, such as contrast enhancement and motion detection, need effective techniques for eliminating rain streaks.

We have focused on rain removal and analysis of images approaches that revolve around adjusting camera settings, in addition to other writers. Because altering the parameters of the camera is not compatible with camcorders, the buyer of a camcorder cannot be linked to previously taken images or videos. On the other hand, [12] for removing raindrops from video, the results of current component-based techniques may be distorted in general.

Furthermore, as an example, rain has predictable consequences, such as rain across along several progressive edges that may damage the pixel. Various image-based demands, such as a simple visual request, object recognition, images selection, astonishing, and picture stitching, depend heavily on the extraction of rotating and scale-invariant point-based contrivances.

However, if the rain lines in images are strongly related to the focus, as shown in Figure 1.1. Rain will be removed from images using a machine learning method. Noises will be lost in the image, and the image's real quality will suffer as a result. It is required to improve image quality by using a new algorithm to enhance and restore image resolution.

Figure1.1: Example of Rainy Images taken by different cameras

From the above images, we can recognize that the raise of object detection will be less and the rate of false detection will be high because of the rain conditions.

1.2 OBJECTIVES

1. To detect the raindrops in the image by utilizing the Deep learning method.

2. To create a graphical user interface to eliminate the noise and raindrops in the images via applying a deep learning algorithm.

3. To show the quality of rain removed image comparison using the histogram to determine the effective method.

4. Feature extraction of images before and after improvement for analysis

1.3 SCOPE

This project will be utilizing machine-learning techniques to locate the rain droplets. Furthermore, noise reduction filters are bilateral and directed filters. Digital images may be made cleaner by using any or both of these effects filters. The rain-removed photos were additionally enhanced using machine learning, as well as a histogram to demonstrate image comparison. The graphical user interface will be developed in Python, and digital images will be utilized to complete the architecture using a novel technique to improve and analyze image resolution.

1.4 CONTRIBUTION

1. To aid in the detection of rain or noise in digital photographs, place a circle on the noise.

2. Capable of removing and analyzing noise and rain from digital photographs taken by multiply cameras

3. Once the noise has been removed, the image quality will be high.

4. The histogram will reflect the characteristics of the original picture and the better picture comparison.

1.5 EXPECTED OUTPUT

Using machine learning to distinguish rain droplets in computerized pictures, image improvement by detecting and eliminating rain in digital photos is possible. The recognized rain droplets in the digital photos will be removed, and the image will be enhanced using deep learning. For comparing techniques, the bilateral filter will be used to compare with the methods, and the accuracy of the enhanced image will be shown in a histogram. For digital photos, machine learning is used to combat the issue of rain droplets.

Image Quality and Target Vision Algorithm Performance Correlation Prior rain detection and removal research has tended to concentrate on picture restoration rather than quantification of the impact of poor image quality on algorithm performance. Rain introduces uncontrollable and irreversible distortions in input images, enough to fool even the most cutting-edge methods for removing raindrops cannot fully correct. We studied the effect of degradation to rain on input images, and it relates to the performance of the target vision system that consumes them. Based on that, we presented a quantitative measure of the correlation between image quality and performance of vision-based algorithms.

1.6 ORGANIZATION OF THESIS

As mentioned earlier, that is dissertation reviews and analyzes the techniques for image quality enhancement and analysis under different rain conditions. Additionally, the performance of methods is evaluated and compared with the dataset. Techniques for area augmentation and analysis are examined in Chapter 2. The research questions and research methodologies of this dissertation are explained in Chapter 3. Additionally, some literature and experiments are reviewed. Chapter 4 discusses the experiments and results. Conclusions and some future directions are then drawn in Chapter 5.

2. RELATED WORK AND BACKGROUND

This chapter examines the most recent advancements in the fields of improvement and analysis. Based on the development of machine learning algorithms, the research looked at how to improve pictures and how to analyze them in different weather conditions.

2.1 QUALITY ASSESSMENT OF IMAGES

Rain may be removed from the atmosphere using a variety of methods. In addition to these studies, they have also conducted a number of others: The removal of rain may be addressed using a problem of picture decomposition based on the morphological principle components of [14]. Low- and high-frequency images may be separated using a bilateral filter for components instead of typical image decomposition algorithms. Using a database and minimalist coding for transfer learning, the HF data is broken down into "rain"-related components. Rain may still be removed, despite the loss of some of the picture's distinctive features.

External vision systems are particularly vulnerable to the harmful effects of rain, which is by far the most prevalent kind of adverse weather. When it rains a lot, rain streaks disperse light, distorting photographs. Image enhancement and object tracking both need the ability to reduce rain streaks. Deep convolutional neural networks (CNNs) have been tasked with this job since they are the most efficient.

Using vocabulary translation and minimalist coding, we were able to remove rain streaks from a single photo using MCA-based image decomposition. There is no need for further training for the suggested method's vocabulary acquisition stage. Picture processing applications like noise removal, augmentation, and painting may benefit from decomposing an image into several semantic components. A new self-learning architecture for picture deconstruction is presented in this work.

The suggested method learns an excessively comprehensive dictionary from the input picture's high spatial frequency components using sparse representation. The team also created a framework for examining the image's dictionary atoms. Decomposition challenges based on sparse representation [16] provided a single-color picture-based rain removal framework. Decomposing an image into its core component (the non-rain region) and the rain streaks it using a dictionary learning and sparse coding approach reveals images with many rainfall noise in a high-frequency part of the picture. The color film may be cleaned of rain splatters using this procedure.

This difficult job has been tackled using single-image approaches, despite the fact that research on video-based methods has proven more fruitful. As an example, a non-local average filtering approach is used in [17] to detect and remove the noise affected by rain and dirt stains from photographs taken via windows, [18] described a similar method based on deep learning. Although this research employs a distinct physiological model, this approach has particular use in mind. We found that this physical model has a limited potential to be used for rain streak elimination, as our subsequent comparisons reveal. In [19], an extended low-rank model, the spatial and temporal correlations acquired by this approach may be used to remove rain from single images and videos.

An established rainy image, originally dissected in [8] into its base layer and detail layer, has been developed in recent research [20–22]. Detailing displays just rain streaks and item characteristics, but the building is still plainly discernible in the background. The sparse code deep learning method is then used to remove rain noise stains from the detail level. De-rained detailed layers are applied on top of the original base layers for the final result. A similar compression method is used in this procedure [23]. This method may be used to remove rain streaks and restore regions that haven't been affected by rain. [24] The detail layer is used to automatically detect rain streaks using a cognitive image decomposition algorithm. Discriminative sparse coding was used by [22]'s authors to clear up a muddled image.

2.2 SINGLE-IMAGE RAIN REMOVAL

For much computational photography and computer vision applications, rainy conditions may hurt performance because of low visibility and visual distraction. [14] – [16]. Image de-raining is essential in these real-world applications for two reasons. As soon as possible, the rain is evacuated from the area in the most efficient manner possible. As a second benefit, it seems as natural as possible once it has been de-rained. This is a difficult issue to solve. The single picture-based rain removal is very ill-posed due to the absence of previous data for both the rain and the backdrop. There are several different ways to de-rain a rain photograph, and the perceived quality of each of these options might vary greatly.

Rain removal from movies is one of the most important image preimages and video surveillance technologies, according to a study [25]. Rain removal is becoming more popular among academics and businesses alike, because of advancements in computer vision technology. They initially analyzed the most common rain removal techniques and grouped them into four groups depending on the rain qualities they were able to take advantage of in the process. After that, some ideas for further study are offered [26]. They utilized a dictionary-learning approach to extract the features of the rain that were not related to the rain's properties. Results from simulations suggest that the proposed approach can eliminate heavy rain from a single picture more effectively than other current methods.

These approaches may easily result in over-or under-smoothing of the rain picture, which might ruin the original image structure or leave a lot of rain behind. Prior rain is captured via low-rank approximation in [27]. It does a good job at estimating rain streaks, but it is prone to mistaking the striped backdrop for rain streaks because of its similar texture. Rain streaks may be more easily distinguished from the backdrop if researchers use techniques like dictionary learning, Gaussian mixture models, and deep learning to concurrently train the priors for both rain and background layers. When it comes to analyzing rain photos, these data-driven solutions do very well. However, their performance would plummet when confronted with a small percentage of the training set's rarer forms of rain photos. A rainy day picture will appear in the actual world.

Because the rain removal artifact degradation levels vary widely from region to region, as shown in Figure 2.1, a de-rained image's overall quality cannot be determined.

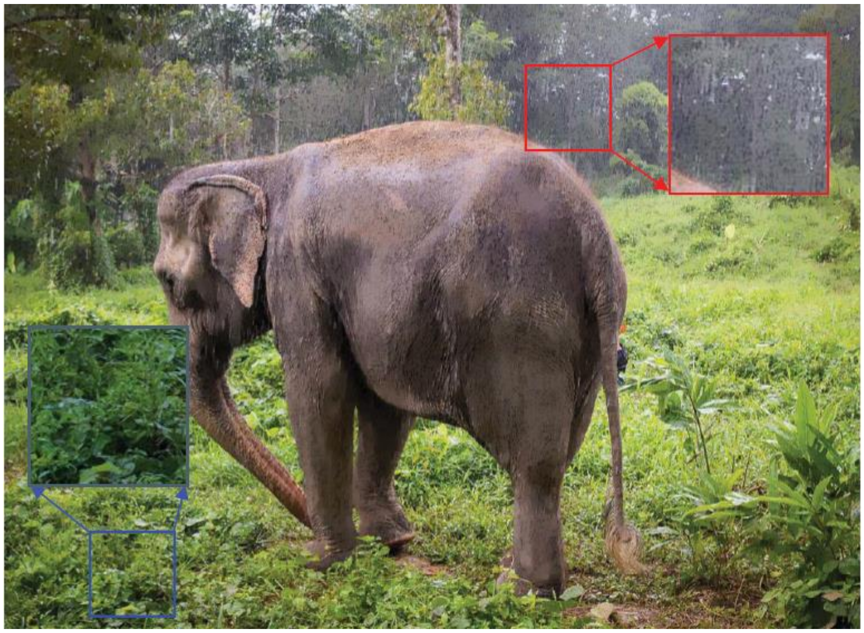

Figure 2.1: The examples of de-raining objects that are not uniform. De-raining outcomes are shown in red and blue bounding boxes, respectively, in this example created by [28]. Because of the unpleasant gaps and streaks in the red bounding box, it is obvious that the quality of the image is poor. The blue bounding box, on the other hand, exhibits excellent quality, with all the leaves retaining their original look.

2.2.1 Methods Based on a Correlation Test and Mathematical Model

Using a mathematical model, an image-to-source relationship may be uncovered using this combination of techniques. Many processes within the camera device are required to capture a picture, and these procedures introduce artifacts into the final image. These artifacts might aid in the identification process in several ways. Due to sensor flaws (dust and noise) and optical aberrations, the approaches under discussion are aimed at assessing such attributes to find a fingerprint for the device. Therefore, the camera's fingerprint is unaffected by the data it analyzes.

To extract the rain component and the rain removal data level from a single rainy image, [29] set out to build an algorithm that was successful at doing so. Screen blend models were used to build an approach for single photograph de-raining that relies on a dictionary learning approach. It was the goal of this project to use highly properties to sparsely approximation patches of two different layers. The non-linear composite of two layers may be accurately separated using discriminative sparse coding. This approach outperforms previous single picture de-raining algorithms on rain photographs examined in the studies. Perceived rainy image quality may be approximated using the metric of structural similarity, as shown in work [30]. Multiscale structural similarity approaches are more flexible than earlier single-scale methods in considering differences in viewing circumstances, according to the researchers. Additionally, an image synthesis technique was devised to assess the characteristics that determine the relative importance of various scales. Their suggested approach is successful by comparing it to other methods.

As part of our effort to cover a wide range of rain scenarios, we first gathered 206 legitimate rain photographs from the Internet, which were taken in a variety of light and angle conditions. There are a few examples of what you may expect to see in Figure 2.2. After removing 1236 raindrops from these actual photos, we provide six typical single-image rain removal algorithms. Lossless compression is utilized to save both the original and de-rained photos to prevent the resulting composite images from being distorted [31].

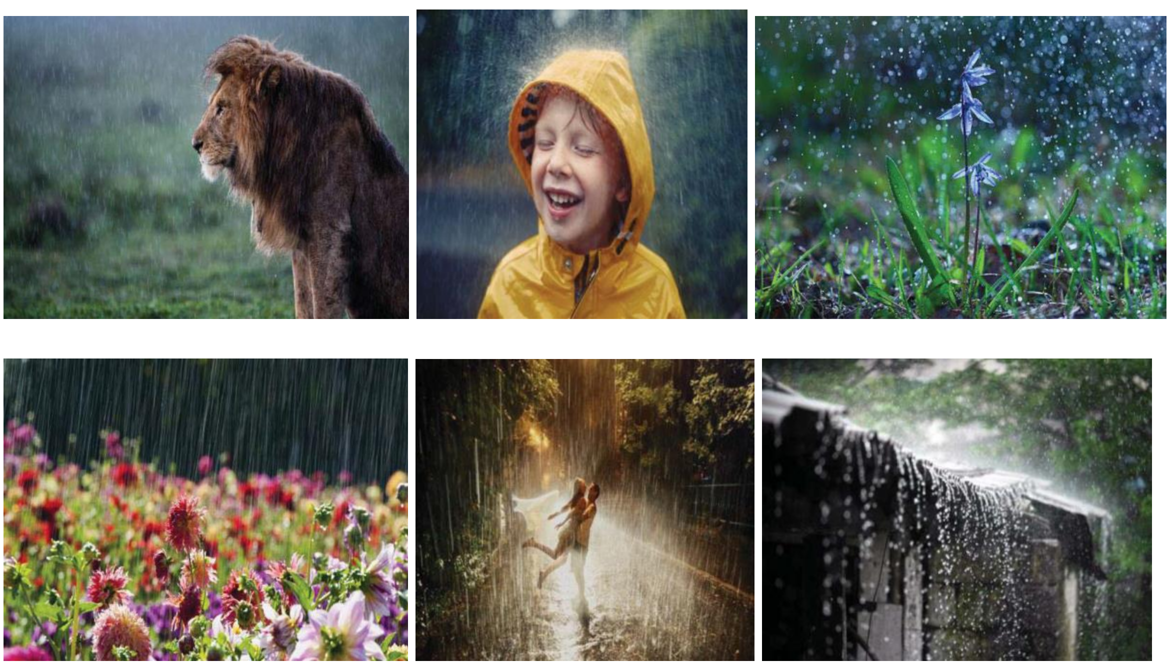

Figure 2.2: Authentic rain images were collected in the de-raining quality assessment database.

For example, the suggested filter, vocabulary learning, low rank approximated, and max value posterior are all included in this paper's investigation of de-raining methods. Their abbreviated names are [14], [29] and [32] – [34] to keep them short. Our analysis relies only on author-supplied code, which is utilized with the default settings, and no further fine-tuning is required. In our rain database, we have six different deluged images for each real rain photograph.

To illustrate, Figure 2.3 shows an obvious comparison of several de-raining techniques. Figures 2.3 (a)-(e) demonstrate how various de-rained photographs have very diverse looks (f). In the following, we put these de-raining techniques through a subjective test to see how well they work.

2.3 DIGITAL CAMERA IDENTIFICATION BASED ON ANALYSIS

Photography is a very popular pastime. Individuals snap pictures and promptly post, those on social media often use Smartphones and digital cameras. Tracing cameras may pose a severe risk to consumers' privacy. For a long time, the issue of tracking digital Lukas has been a hot topic. The sensor of a camera is recognized when it is traced. The term "hardware entry" refers to the process of looking for distinguishing characteristics to identify a camera. As stated in [36], [37], each camera leaves behind a "camera fingerprint" that may be used to identify and track the camera. Pixel artifacts caused by sensor flaws or optical errors are the most common explanations for this kind of fingerprint.

In [37] introduced one of the most well-known and up-to-date camera recognition algorithms. It is found that for each camera, which functions as a unique camera fingerprint, may be detected. A de-noise filter averages the noise from several pictures taken by different cameras to arrive at this sensor pattern noise. A wavelet-based de-noising filter, according to the authors, is the most efficient method for removing noise from pictures, and the results of an experiment indicate this. Such a method is successful, and the camera's identification is excellent. The wavelet-based de-noising filter, on the other hand, takes a long time to apply [38].

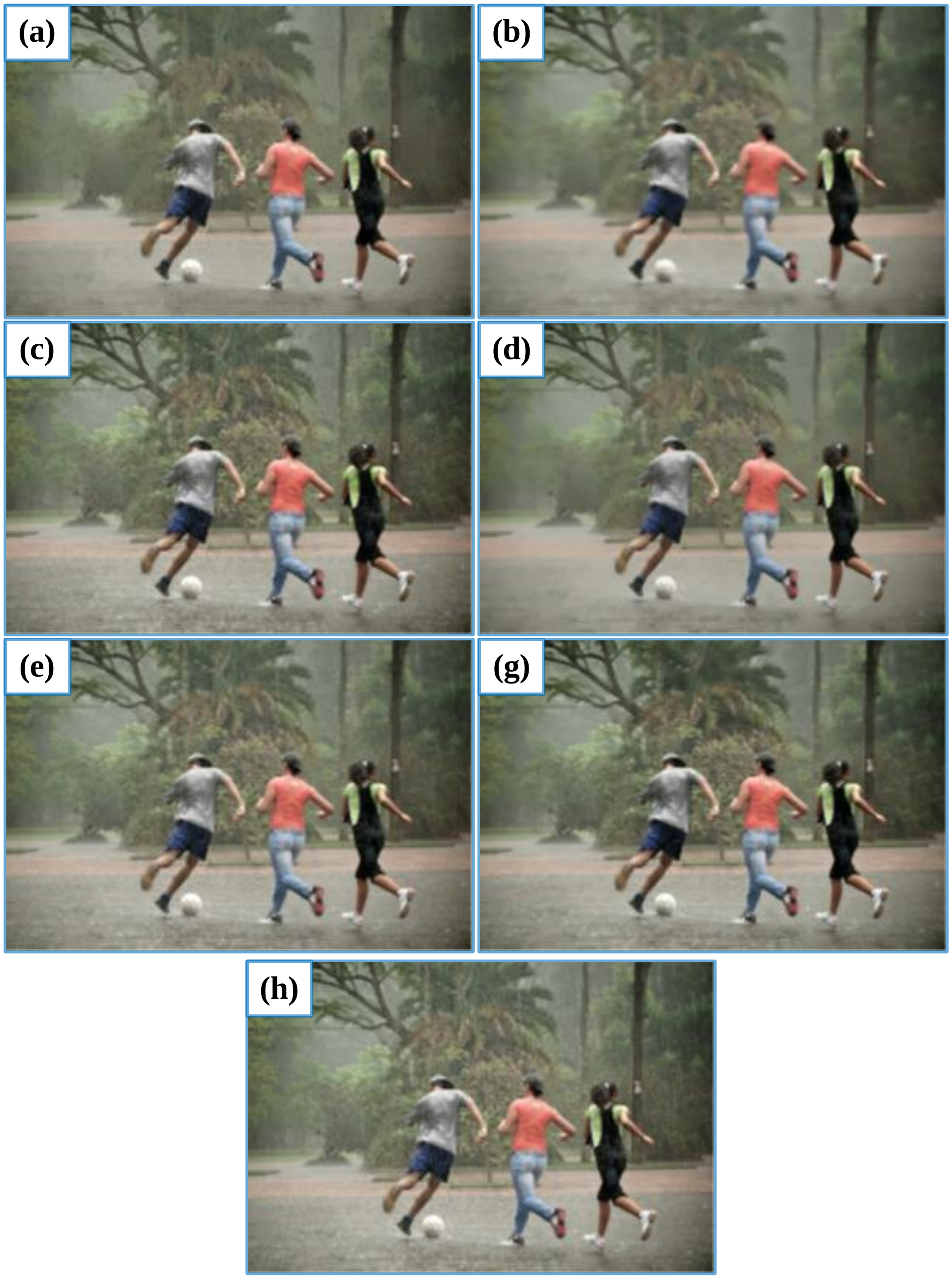

Figure 2.3: An example of how various algorithms generate de-rained photos. Where (a) is [14], (b) [29], (c) [32], (d) [33], (e) [33], (f) [34], and (g) is input image.

PSNR approach developed in [39] is a quick camera tracing method. In comparison to [37], the method is faster, but the classification accuracy is lower Photo Responses Non-Uniformity (PRNU) patterns may be managed using the k-means algorithm, according to [40]. Correlation is used to compare patterns, and the k-means method is used to organize them. As a result, a cluster of identical patterns is assumed to be from the same camera. A database of 500 photographs was used to perform the tests. The TPR (true positive rate) of the images in a cluster was 98 percent, indicating that they belonged to the same camera. [41]. Also introduces the concept of fingerprint clustering.

JPEG compression is analyzed in [42]. JPEG loss compression is well known for its noise, which affects groups of pixels. According to experts, JPEG compression has distinctive artifacts that might be utilized as a means of detecting the original image (source image).

This approach takes a long time to complete because of the de-noising of the pictures. [43] Camera recognition was examined in terms of dead pixels, traps, points/hot points, and cluster flaws [44]. In certain situations, hot and dead pixels may be used to identify a camera's sensor based on its faulty pixel pattern, according to experimental data.

A comparable approach is provided in [45]. An image's sensor fingerprint is regarded to be a white noise that is present in the picture. To lower the False Positive Rate (FPR) of camera recognition, authors recommend employing the correlation to circular correlation standard as a test statistic. However, unlike [44], the suggested approach is tested on "fragments" of pictures, rather than "whole" photographs. 95 % of the original 256x256px images were recognized as positive and the final recognition accuracy was 99%. (512x512px).

An identifying approach for cameras is mentioned in [43]. It is in the same vein as [44] to figure out a camera's fingerprint. The Peak to Method Achieves Ratio is used for evaluation. Images used in the experiments are taken from the popular photo-sharing website Flickr. More than a million photos from 6896 cameras representing 150 models were analyzed in the tests. Support vector machines (SVM) and data fusion approaches are used to tackle the same issue in [46].

2.4 RAIN MODELS

Researchers in a variety of domains, including architecture, communications, and atmospheric sciences [6], [7], have been studying rain phenomena since the 19th century. A foundation was laid for future pioneering work on rain removal, which examined and simulated several aspects of rainfall characteristics [6]. The researchers themselves were not intended to address the computer vision challenge of eliminating rain. The initial in-depth and thorough investigations of rain's effect on collected images, which were utilized as the major literature in many rain removal algorithms, developed dynamic and spectroscopic models. However, dynamic rain models take into account changes in rainfall properties over time, which are not taken into account by photometric models. The appearance of rain is affected by a variety of factors, including the terminal velocity of raindrops, their diameter, and their volumetric distribution in space, as well as the camera's settings and the ambient light. Because of this, the study's focus was on employing camera settings altered while recording a rain scene to identify and eliminate rain.

2.4.1 Rain Properties

Shape: Wind tunnel experiments sparked the idea of experimenting with raindrop morphologies by dropping them from a high altitude. The field of atmospheric sciences has documented several experiments and theoretical studies on the structure of raindrops as they fall at their terminal velocities dating back to 1883 [56]. Oscillations, a characteristic depicting a rapid change in the shape of raindrops that results in complicated intensity fluctuations in a rain image, occur. Slight variations in light reflection and refraction patterns at individual raindrops may affect an image's brightness or even its overall appearance [57]. Fortunately, oscillations have little effect on rain pictures since rain falls at its terminal velocity in the sky. Figure 2.4 [6] depicts the equilibrium forms of raindrops of different sizes.

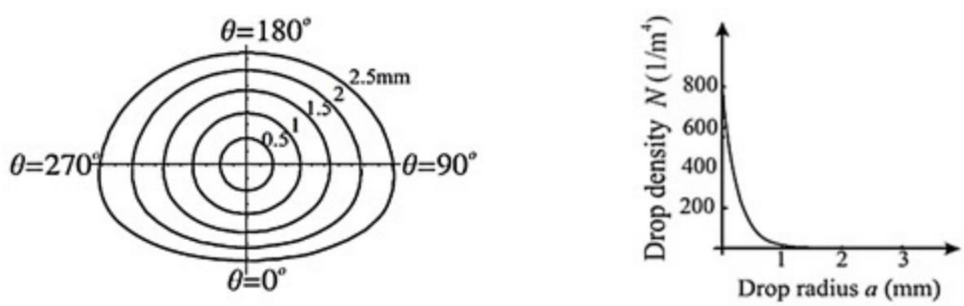

Figure 2.4: The concentration of raindrops of different sizes and their shapes [6].

The diameter of rain droplets with spherical shapes is often smaller, whereas the diameter of raindrops with oblate spheroid shapes is typically greater, which may be defined as a 10th order cosine distortion of a sphere. It is possible to express the raindrop's diameter, r (q), using Equation (2.1) below.

$$r(\emptyset) = a(1 + \sum_{n=1}^{10} c_n COS(n\emptyset)), \qquad (2.1)$$

There are 10 raindrop coefficients (c1-10) dependent on the raindrop's diameter (a), and one additional coefficient (q) dependent on its polar angle of elevation (a) [6]. q = 0 is located at the raindrop's egress path.

Size: Raindrops may be as little as 0.1 mm in diameter and as large as 10 mm in diameter. When it comes to raindrop size distribution, the Marshall-Palmer distribution [58] is the most often utilized method.

$$N(a) = 8 * 10^6 e^{-8200*h^{-0.21}a}, \qquad (2.2)$$

N (a) is the total number of rainfall per cubic meter that fall within the restricted range of fluctuation (a, a + da) when calculating the number of droplets per cubic meter of water. Figure 2.4 depicts the frequency distribution of a typical 30 mm per hour rate. Most raindrops have a diameter of less than 1 mm, as seen in the image. On the other hand, as seen in Figure 2.4, the lowest distortion of such typical raindrop sizes may be approximated or described as a spherical form.

Velocity: After falling for at least 12 meters without hitting any obstacles, the final velocity of fast-falling raindrops might be between 5 and 9 m=s [59]. The following empirical formula is used to calculate the ultimate velocity v of a raindrop of a specific radius:

$$V \rightarrow \; = 200\sqrt{a} \qquad (2.3)$$

2.4.2 Rain Dynamic Model

Rain and storm pixels in a movie may be converted into a binary array with a value of one or zero [6]. The volumetric positions of rain droplets may be calculated by connecting the spatial and temporal information in consecutive borders of the video. Detecting rain and its direction are as simple as seeing pixels that have a strong correlation in time [6]. This is because raindrops move in a straight path with a constant velocity. The distance between rain pixels in one frame and the next in the picture plane is always the same in the direction of the rain [6].

2.4.3 Rain Photometric Model

The refraction of incoming radiation from the surroundings by raindrops is the primary reason for their complicated look under intense illumination. High-velocity raindrops appear like rainfall stripes to the visual system and to cameras due to motion artifacts [6]. The projected raindrop duration onto a pixel (1.18 ms) is discovered to be the origin of this phenomenon, which is substantially less than the standard video camera exposure time (30 ms) [6]. After a brief exposure time (1 ms), the rain droplets look stationary, bigger, and non-transparent, according to experiments done by authors. In other words, raindrops' motion-blurring effect causes rain streaks to look transparent to humans and cameras alike. Lighting effects on rain streaks may be influenced by a variety of camera settings, including the amount of ambient light brightness and exposure duration [6]. It has been shown that, in terms of quantitative variables of distinct RGB constituents' intensity levels, they are equivalent for the same region of the image [17].

2.5 ANALYSIS OF DIFFERENT IMAGE FEATURES

The primary function of a camera, as everyone knows, is to take images, but the idea of picturing has developed year after year since the first black and white cameras. The goal is to capture a 2D photo after removing the obscurity from the item with enough light. The spatial domain positions determine the coordinates in a two-dimensional image [47]. It is important to understand that a digital picture is made up of a limited number of components, each with its amplitude and position, which are referred to as image elements or image pixels. As a result, the image pixel is the unit for representing digital picture components. As a result, digital image processing has a broad range of applications [48].

The number of pixels in digital imaging is referred to as picture resolution in image processing. The smaller the pixel size is, the greater the resolution and the clearer the item in the picture will be with any particular lens setting. More pixels may be present in images with lower pixel sizes. The quantity of information in a picture is proportional to the number of pixels. Pixel resolution is represented by a pair of positive real integers, the first of which indicates the picture width or image pixels' divisions and the second of which indicates the image height or the number of pixel rows, such as 7680 by 6876. A picture with a width of pixel resolution and a height of

2048 pixels does have a total of 1024 2048 = 2097152 pixels [47], [49]. Views from a Variety of Cameras.

2.5.1 Digital Cameras and Cellphone Cameras

It would take a book-length essay to discuss the first cameras, so let's start with digital ones: Eastman Kodak Company engineer S. Sasson invented the first electronic camera employing a charge-coupled device image sensor in 1975 [36]. It utilized tubes in the early days, but later cameras digitized the signal and utilized technology to accomplish their aims. Sharp debuted the J-SH04 J-Phone, the world's first digital camera phone, in Japan in 2000, as academics and businesses began to embrace electronic and digital technology, and it was just a matter of time. Throughout the mid-2000s, high-end mobile phones were equipped with built-in digital cameras. Most smartphones will have a camera integrated in this year.

The global rise in mobile phone usage has unmistakably altered the global technological environment. There are strong expectations that, after the success of text messaging, mobile phone providers would be able to benefit even more from this massive market (especially in Europe and Asia). According to research conducted in Japan, camera phones account for more than half of the mobile phone market, with major operators such as J-Phone stating that more than 70% of clients subscribe to MMS (Multimedia Messaging Service). As the camera on a mobile phone becomes more exact, its value rises [50].

The flashlight became important in mobile phone cameras since it is used in cameras (see Figure 2.4).

Figure 2.4: Different camera types with a flashlight (continue figure)

It is crucial not to overlook the flashlight included in both cameras and mobile phones. Although many cameras come with an electric flash, the majority of them need a separate device. These devices, known as an on-camera flash, fit onto a bracket on top of the camera called a shoe. There are numerous different types of flashes, but practically all of them may be divided into three categories: TTL auto flash, non-TTL auto flash, and manual [47].

2.5.2 Rain Disruption on Images

According to the preceding section's description of rain phenomena, the appearance of raindrops in an outdoor photograph is reliant on the distance z between the camera lens and the volume of raindrops, in addition to the brightness of the surroundings. That is why it is necessary to deal with at least two different types of rain appearance, one in the front and one in the background. When designing our suggested network to eliminate rain interruption from photos, we may make use of this simplified look of rain on photographs. Foreground rain is removed in the first step, while background rain is removed in the second step, which removes the haziness in the background produced by the accumulation of small raindrops in the foreground and the background.

2.6 FEATURE EXTRACTION

Gray Level Co-occurrence Matrix (GLCM**):** A GLCM for Feature Extraction: The brightness, correlations, energy, and homogeneity values from Robert Haralick's GrayLevel Co-occurrence Matrix are employed as the initial texture characteristics. The contrast shows local changes in the co-occurrence of gray-level matrices in the matrix. Then, the correlation displays the likelihood of the linked pixels. GLCM's combination of squared dements represents the energy. The homogeneity or second-moment angle are other names for it. It is at this point that homogeneity is measured, which is how close an element's distribution in the GLCM is to its diagonal values. GLCM value is only obtained when a co-occurrence matrix has been constructed (CM) [51].

Discrete Wavelet Transform: The objective of extracting texture characteristics is to gather new variables from the matrix image that have specific information and then use these variables to classify the image. A Gaussian distribution has been used to represent the wavelet coefficient distribution. As DWT characteristics, approximations, horizontal detail, vertical detail, and diagonal detail were dissected; a simple statistical methodology employing the Standard Deviation coefficient was employed (1) and (2) show how the standard deviation may be used to assess the average contrast of an image region (2). When contrast is strong, a rough texture can be seen, and when contrast is low, a smooth one may be seen [51].

2.7 EARLY RAIN REMOVAL APPROACHES

Reconstructing the rain-disrupted scene using just one photograph is a more difficult process than using video. The video technique and the single-image approach to rain removal are analyzed in-depth in this article. Rain removal is often thought to be solvable using standard methods that don't use deep learning, the radiometric method of the input image or, video or the model that changes throughout time for a motion. We'll move on to a more difficult topic after studying video rain removal algorithms: eradicating rain disruption from an image. As part of the background research.

2.7.1 Rain Removal from Video

Moving rain streaks may create a blurred image, therefore to avoid this, the authors suggest adjusting the camera's G function or extending the exposure duration T, as described in the prior section [6]. This method is best suited for a static or basic backdrop [63]. For removing rain from a movie, the authors recommend another technique that involves first detecting and then segmenting the pixels of single rain streaks within successive scenes, using a model of the look of rain streaks. Although this technique may work in a dynamic scenario, it may also be used in a plain backdrop scene without any texture.

Several video rain detection methods have been suggested. Figure 2.5 depicts the whole picture. It is possible to eliminate the rain effect in a moving picture using noise-filtering methods in the No Explicit Detection approach, however, this method fails to detect rain when the scene is dynamic. Images containing rain are divided into the rain and rain-free pixels using the Per-Pixel Detection method [17], although due to the rain's temporal features, some images will be misclassified. Each video frame is subjected to PBD, which utilizes the rain photometric method to identify rain areas [6]. Due to the rigorous thresholding approach of classifying rain pixels, this algorithm may miss a rain-free zone with other moving objects. An estimation of rain's energy about other objects' Spatio-temporal frequency domain is used [63] to identify rain in video. Only global patterns of precipitation are taken into account in the frequency domain model, which makes it difficult to identify rain and then remove it [63].

Once the rain has been detected, it is possible to remove the rain pixels or regions. It is possible to remove rain from video using two basic techniques, both of which are based on the de-raining approach in the frequency domain. As seen in Figure 2.5, both strategies use video rain removal algorithms.

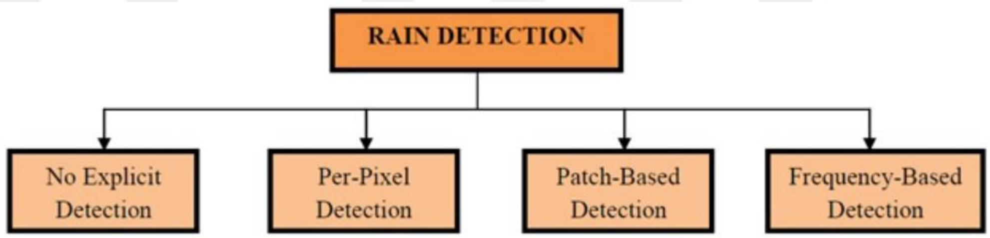

Figure 2.5: How to classify rain in video footage [16]: a taxonomy.

Video rain identification and removal may also be improved by using rain's attributes in both the temporal and spatial domains, as well as the frequency domain. A rain pixel's amplitude information may be recognized in successive frames of the video, but it can be conducted separately, utilizing its neighbors in the video frame as replacements [17]. As a result, the substituted rain pixels tend to seem darker than their original counterparts [69]. Based on spatial frequency, identification is used to remove rain, as seen in Figure 2.6 (b).

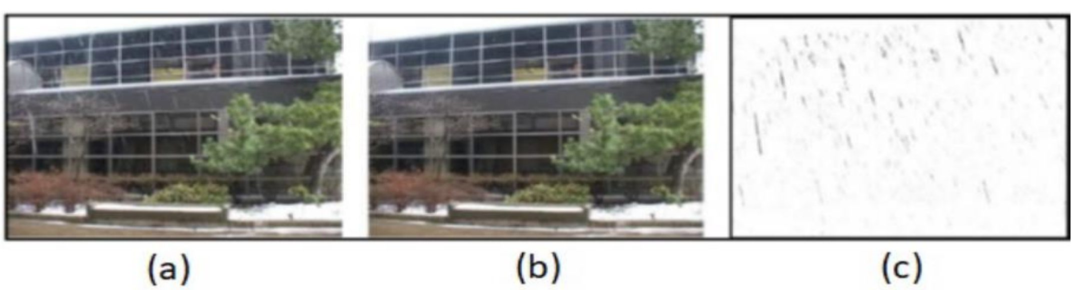

Figure 2.6: After subtracting successive frames from a video series (a), the rain detection findings (c) of a frosted glass structure with multiple vertical and horizontal edges and (b). Near windows frames and plants, there are some false alarms (reproduced from [16]).

3. METHODOLOGY

The third chapter contains a description of the developed computational technique and algorithmic support for extracting data from interacting information processes for processing and analyzing visual data in image processing and analysis under different weather conditions, the recommended system is shown in Figure 3.1.

If it is not possible to obtain data that directly describes the state of the bad quality of the rain. The solution to this problem was performed using machine learning, and the developed algorithmic software made it possible to increase the accuracy of images taken under the rain, feature extraction, and analysis. This chapter also outlines a deep neural network learning technique and how it can be used to enhance the image under bad weather images. Image quality enhancement will increase the accuracy of objects detection under rainy weather, as well as this kind of process, plays an important role in security cameras and object predication.

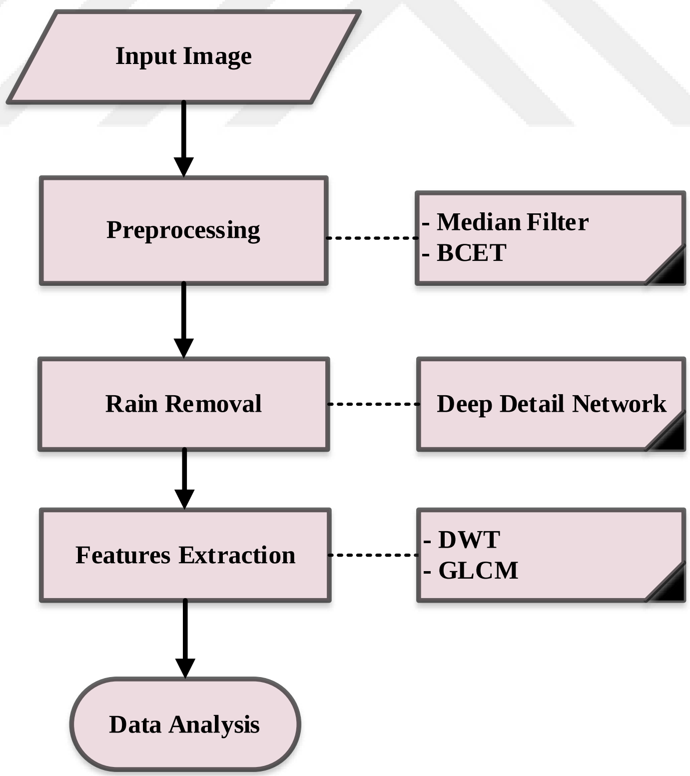

Figure 3.1: The flowchart of recommended system

3.1 IMPROVING THE CONTRAST OF IMAGES

The main function of preprocessing is to improve the images quality of the local studied area (taken by different cameras) under rainy weather. Additionally, a shape is created suitable for more processing by machine vision systems or humans. Additionally, specific image parameters are improved by preprocessing, such as the following [75]:

Pre-major processing is used to improve the features of photos taken in wet conditions. Removes noise and changes undesirable areas of the backdrop while also increasing many picture metrics in this stage (for example, PSNR). In the initial step of noise reduction, a median filter with picture set processing changes was used. Weights were also determined based on the distance between consecutive rainy photos. It is used as a starting point to formulate the median filter using frequency distribution [70]. An adaptive weight modification based on local pictures improves noise reduction quality [71]. The histogram construction approach that takes weight into account is shown in Figure 3.2.

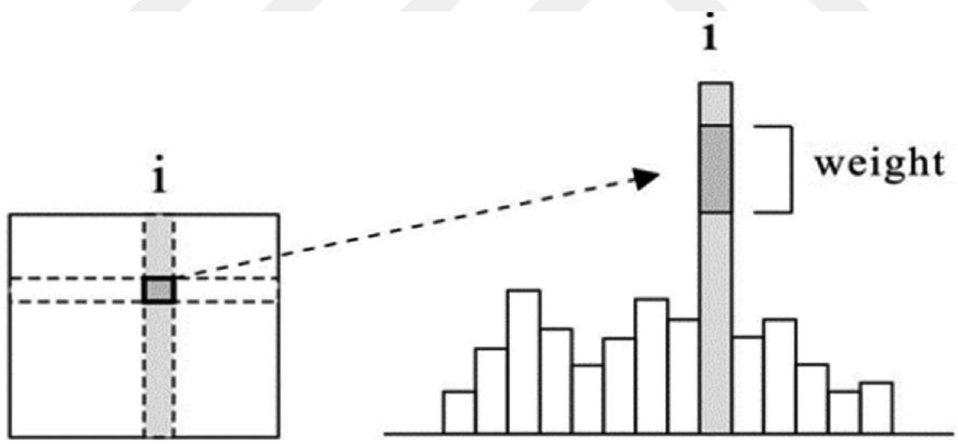

Figure 3.2: Scheme of weighted median filter histogram formation

Because the CT scans might vary in contrast enhancement, the Multi-Scale Retinex (MSR) modifications were used, in which the wavelet decomposition was employed to speed up the computations [72] to lighting and intensity adjustments in the AoI. The ability to compute textural features and analyze the structural aspects of the lesion is aided by a brighter representation in the lung region. Then, we apply adaptive contrast enhancement based on the modified function (described in [73]) to get a better signal-to-noise ratio. Consequently, improving the purity of the original MRI images. However, the main obstacle is poor image quality to extract, analyze, recognize efficiently, and quantitively measure parameters. Due to some imperfect imaging process characteristics, medical images are usually made impure by impulsive, multiplicative, or additive noise. The authors in [74] proposed using Contrast Balance Enhancement Technique (CBET) technology to get better contrast of highlighting the interest area. It is worth mentioning, the area of interest requires contrast during medical imaging.

The articles [75] described the Contrast Limited Adaptive Histogram Equalization (CLAHE) technique. The method aimed to improve the characteristics and determine better medical image characteristics, leading to adequate diagnosis. For image enhancement, the is another interesting approach which is Unsharp masking. The goal is to improve edges and detail; however, using a high-frequency emitter makes this technique very sensitive to noise. In his work [76], the author showed that invariant contour transform (STICT) improves the MRI scans quality.

In the suggested method, we use the algorithm (BCET). Image contrast can be compressed or stretched without altering the shape of the histogram of the input image (I_{old}). In the solution, a parabolic function is obtained from the input image. The parabolic function equation is written as follows:

$$I_{New} = a \cdot (I_{old} - b)^2 + c \ . \tag{3.1}$$

The coefficients a is derived from the input. Whereas b and c are determined from the minimum output image value (I_{New}), the maximum output image value. The following equation calculates the value of the average output image:

$$b = \frac{h^2 \cdot (E-L) - s \cdot (H-L) + l^2 \cdot (H-E)}{2 \cdot [h \cdot (E-L) - e \cdot (H-L) + l \cdot (H-E)]} \ , \tag{3.2}$$

$$a = \frac{H-L}{(h-l)(h+l-2b)} \ , \tag{3.3}$$

$$c = L - a(l-b)^2 \ , \tag{3.4}$$

$$s = \frac{1}{N} \sum_{i=1}^{N} I_{Old}^2(i) \ . \tag{3.5}$$

where H and L are the source image's highest and lowest values, respectively. In addition, H and L are the produced image's maximum and lowest values. The average of the input picture is denoted by s, whereas the mean - squared summation of the image pixels is denoted by e. All channels have their minimum and maximum values set to 0 and 255, respectively. And the average value is 120. Figures for each color channel's mean value are presented in the Table 3.1 and 3.2.

Table 3.1: RGB channels of image before applying BCET

SOURCE-IMAGE	MIN	MAX	MEAN
RED channel	1	256	45
GREEN channel	1	256	59.36
BLUE channel	1	256	42.7

The results below demonstrate that after using the BCET solution, the mean values of all the channels have been balanced. An example is shown in Figure 3.3.

Table 3.2: RGB channels of image after applying BCET

SOURCE-IMAGE	MIN	MAX	MEAN
RED channel	1	256	119
GREEN channel	1	256	119.8
BLUE channel	1	256	120

Figure 3.3: Example of color image processing with BCET

3.2 IMAGE DE-RAINING

In video and image processing, rain streaks may lead to a significant loss in performance. Many researchers in computer vision and image processing have been focusing on rain removal from video and images lately, and many algorithms have been presented to accomplish this goal.

Our goal is to improve the rainy image from rain streak on single images to increase the object detection results. Hence, obtain a real-time-efficient system that can work even in bad weather conditions (like rainy days) and help visually impaired people in their daily lives.

Following the huge success of deep neural networks, a deep architecture [4] for rain removal was investigated. A deep convolutional neural network is used in this system (CNN). The deep residual network (Res-Net) is essential to this network's operation [80], [81]. To simplify the learning process, Res-Net aims to change the mapping form. Deep convolutional neural networks were used to remove rain streaks from a single picture, realizing the same concept.

Since the outcome of the rain removal process will be used in object detection, we needed a method that preserves the quality of the image. For this purpose, we studied an algorithm with the highest qualitative and quantitative measures called Deep Detail Network (DDN). The general framework of the algorithm is presented in Figure 3.4.

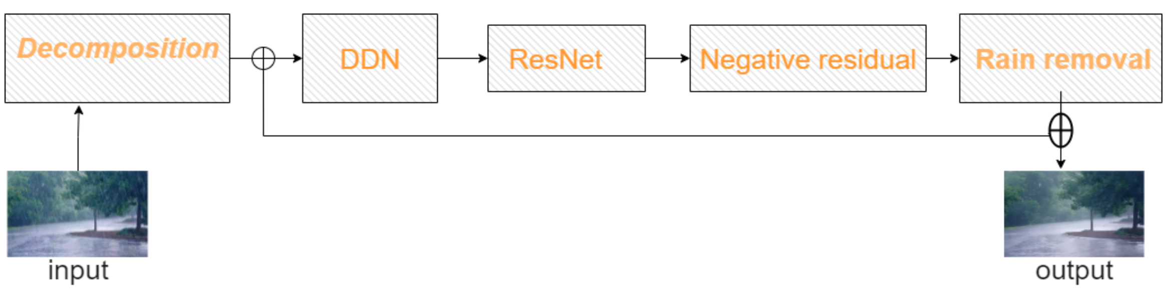

Figure 3.4: A picture depicting the structure of the rain reduction algorithm.

3.2.1. General idea (negative residual mapping and deep detail structure)

The proposed work uses the characteristic of rain streaks to enhance the network result and give some sort of bias. The authors first applied a standard deep convolutional neural network, but the outcomes of the networks were not satisfying as seen in images in Figure 3.5.

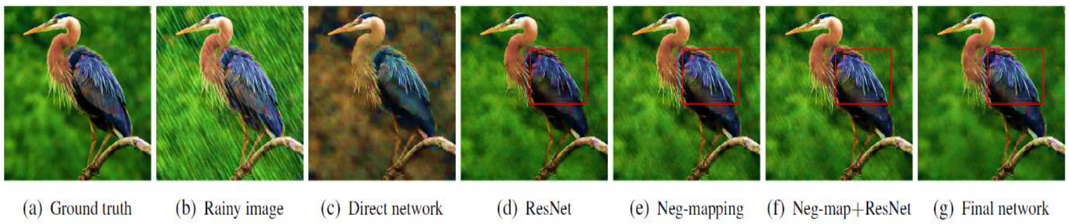

Figure 3.5: Effects of diverse network architectures on the de-raining process [4].

We can see that the direct network in Figure 3.5.c shows that image color space changed, so the network not only removed the rain but also changed the range of pixels (colors) in the image. The researcher briefly explains this phenomenon to make it simple, let us think that there is a normalization of the images X and Y to [0-1] with D pixels. The regression function between these images' maps from $[0,1]^D$ to $[0,1]^D$ and must be obtained. This means to map a range covering the whole pixel values making it difficult learning how to do regressions correctly A gradient that disappears when regularization techniques like batch normalization are employed might be an issue when training a deep network directly on pictures. (Figure 3.6).

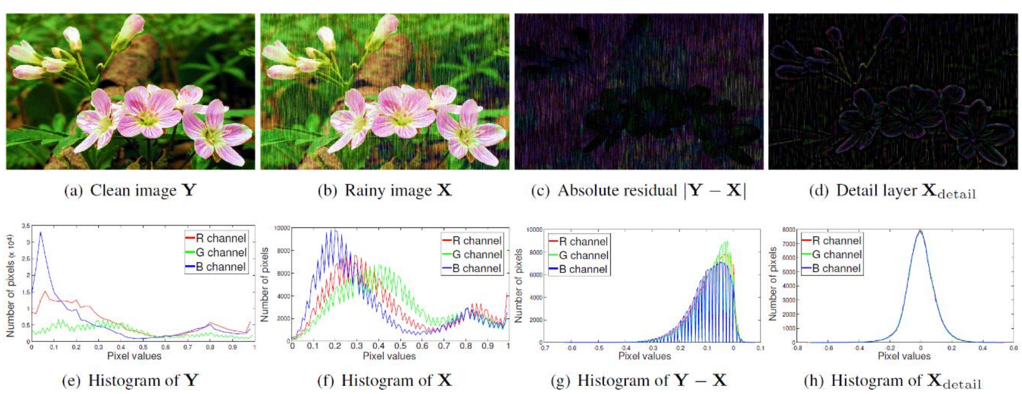

Figure 3.6: Range reduction, detail layer, and negative residual sparsity. |Y-X| to improve the visualization [80].

As network intelligence improves, it becomes more important to reduce the amount of space required for a mapping run. A key run reduction of pixel units (see Figure 3.6 (e) and (g)) with regards to the improved image, which is the residue of the windy data Y-X. In this way, the residual may be supplied to the network, which will help with the mapping process. As illustrated in Figure 3.3, the residual was used as a yield for the parameter layers.

This simple skip association may result in lossless data for the whole ears, which is useful for analyzing the final de-rained image. Figure 3.6 (g) shows that Y X might be negative in most circumstances since rain appears as white streaks in photographs. Alluded to "to this as negative leftover mapping" in this manner (neg-mapping for brief). A revised goal that brings the concept together:

$$L = ||h(Xi\ ) + Xi - \ Yi\ ||_F^2 \tag{3.7}$$

In Figure 3.6 (d), the de-raining produced from the common Res-Net model [81], is modified to suit the image regression weakness. Rain streaks are removed in the original Res-Net structure while blurring the bird's feathers. However, Figure 3.6(e) shows the details of both the color and object by setting in the neg-mapping with no use of the Res-Net structure. This is caused by the lossless information being in direct propagation and updating parameters with the help of the skip connection. In addition, neg-mapping has fewer testing errors and training than Res-Net.

In the long run, it can be seen within the produced pictures that not all rain streaks disappeared. This infers that rain streaks and protest points of interest are not completely recognized by the show. This spurs the taking after improvements of the neg-mapping thought.

Deeper architectures could grow the capacity and flexibility to explore and model the image features [82], a Res-Net [81] structure implemented with neg-mapping for better differences between the rain streaks from other details of objects. In this structure, the input information is guaranteed, and its propagation happens by all parameter layers, training the network. Yet, Figure 3.6 (f) shows that slight rain streaks remain in the produced image even with the combination of both. So, the source Res-Net approach is different from the detail layer input to the parameter layers. Thus, the first model is:

$$X = \ X_{detail} + \ X_{base} \tag{3.8}$$

where the terms "detail" and "base" refer to the various levels of information. To get the detail layer Xdetail, first, apply a low-pass filter to X and then multiply by X-Xbase to produce the base layer. Base layer omission leaves just rain stripes and feature patterns in the detail level, as seen by Figure 3.6 d and e. (h). As a result, the detail layer's sparsity is greater than the image's since most areas are near to zero in the detail layer. There are no existing techniques for de-raining that leverage this sparsity, while the foundation of deep learning is pure.

As an example, Figure 3.6 (c) illustrates that "both the detail layer and the neg-mapping residual exhibit essential extend lowering" may be seen. Thus, the viable mapping is generated by smaller subsets of $[0, 1]^D$ to$[0, 1]^D$. [80] As a result of the shrinking network space, network execution should be advanced.

This helps the combination of the detail layer $Xdetail$ with the suggested neg-mapping Y − X because there is an input to the Res-Net parameter layers. Scholars training the detail layer network name this “deep detail network.” In Figure 3.5(g), the last output is produced from the combination of the designed deep detail network and neg-mapping. This output helps achieve a good convergence rate and a good structure similarity index (SSIM), yet there is a cleaner visual de-raining influence.

3.2.2 Network architecture

The de-rain outline input could be a blustery picture X and the yield is close to the clean picture Y. Thus, they characterized the objective work to be:

$$L = \sum_{i=1}^{N} \quad ||f(X_{i,detail}, W, b) + Xi - \ Yi \ ||_F^2 \tag{3.9}$$

where N is and $f(\cdot)$ they are Res-Net number of training images respectively. Also, W and b are two network factors that must be known. X_{detail}, guided filtering is utilized [88] as a low-pass filter for the division of X into detail and base and layers. The removal of the image indexing and basic network structure is:

$$X_{detail}^{0} = \ X - \ X_{base},$$

$$X_{detail}^{1} = \sigma(BN(\ W^1 * \ X_{detail}^{0} - \ b^1)),$$

$$X_{detail}^{2l} = \sigma(BN\big(\ W^{2l} * \ X_{detail}^{2l-1} - \ b^{2l}\big)),$$

$$X_{detail}^{2l+1} = \sigma\left(BN\big(\ W^{2l+1} * \ X_{detail}^{2l} - \ b^{2l+1}\big)\right) + X_{detail}^{2l-1},$$

$$Y_{approx} = BN(W^L * X_{detail}^{L-1} - b^L) + X, \tag{3.10}$$

Here $l = 1, \ldots, \frac{L-2}{2}$ with L are all layers, $*$ is the operation of convolution, W is the weights and b biases. $BN(\cdot)$ Represents the batch normalization for the alleviation of the internal covariate shift [89]. σ $(\cdot)$ stands for the Rectified Linear Unit (Re-LU) for non-linearity. Here, all no pooling activities are detached for the preservation of spatial information.

In the first layer, size filters $c \times s_1 \times s_1 \times a_1$ are utilized for the generation of a_1 feature maps; s is filter size while c is the image channels, e.g., $c = 1$ for gray-scale and $c = 3$ for a color image. The layer uses filters of size $a_2 \times s_3 \times s_3 \times c$ for the estimation of the negative residual. The direct addition of the estimated residual to the rainy image X produces the de-rained image.

3.3 FEATURE EXTRACTION

Color characteristics, texture, shape, and contrast are all examples of quantitative information that may be extracted from a picture. In order to compare the picture quality before and after rainy weather processing, we employed DWT to extract wavelet coefficients and GLCM to extract statistical features.

3.3.1 Discrete Wavelet Transform (DWT)

A picture obtained by a variety of cameras during wet conditions was processed using a wavelet algorithm. DWT, a strong feature extraction tool, is used here. The rainy images were utilized to obtain the coefficient of wavelets. The enhancement was made easier because of the wavelet's ability to locate frequency information inside a signal function.

LL (low–low), HL (high–low), LH (low–high), and HH (high–high) sub-bands were created using two-level wavelet decomposition of the Region of Interest (ROI). It is possible to approximate low and high-level frequency content in a picture using the 2D level decomposition method [77]. The low-frequency pictures are represented by LL1 and LL2 wavelet approximations; these are the first and second layers of wavelet approximation. LH1, HL1, HH1, LH2, HL2, and HH2 make up the high-frequency portion of the pictures and represent the first and second levels of horizontal, vertical, and diagonal directions, respectively. A low-level picture, LL1, is utilized to approximate the original image, which is then divided into second- and third-level approximations and details. We kept going back and forth between these steps until we got to the appropriate pixel count.

The pictures were split into spatial frequency components using a 2D discrete wavelet transform, and while HL sub-bands had greater performance than LL, we employed both LL and HL for better analysis that defines image-text characteristics [77]. In order to study the various frequency components and each component, we used a resolution that was appropriate to its scale.

$$DWT\ p(s) = \{d_{i,j} = \sum \quad p(s) * h * i(s - 2ij)\ d_{i,j} = \sum \quad p(s) * g * i(s - 2ij) \tag{3.6}$$

Coefficient $d_{i,j}$ is the component property in signal p(s) corresponding to the wavelet function, while $p_{i,j}$ relates to the estimated components. There are two types of filter coefficients in this equation: high-pass and low-pass, which are represented by h(s) and g(s), respectively; wavelet scale and translation factors are represented by i and j.

3.3.2 Grey Level Co-occurrence Matrix (GLCM)

Images may be analyzed statistically using the GLCM method. Second-order statistics may be derived from the connection between two pixels in the same offset. Gray-level-I and gray-level-j pixels are counted horizontally when using the default configuration for GLCM, which computes the number of times each pixel appears next to the other. Reference and neighbor pixels are used interchangeably in this context. However, by adjusting the offset parameter's value, other pixel spatial correlations may be found. Row and column offsets of [0 1] represent the distance between the reference pixel and its neighbor. There are a number of different names for this matrix, including a two-dimensional GLCM matrix of joint probabilities among pixels pairs. [78].

An example of how the GLCM computes GLCM (1, 1) values is shown in Figure 3.7. Because the picture only has one instance of two horizontally adjacent pixels with a value of 1, we can tell that this point in the matrix carries the value 1. Because the picture has two instances of pixels with values 1 and 2, we can observe that GLCM (1, 2) contains the value 2 as well. There are several uses for the co-occurrence matrix, such as biological imaging [79].

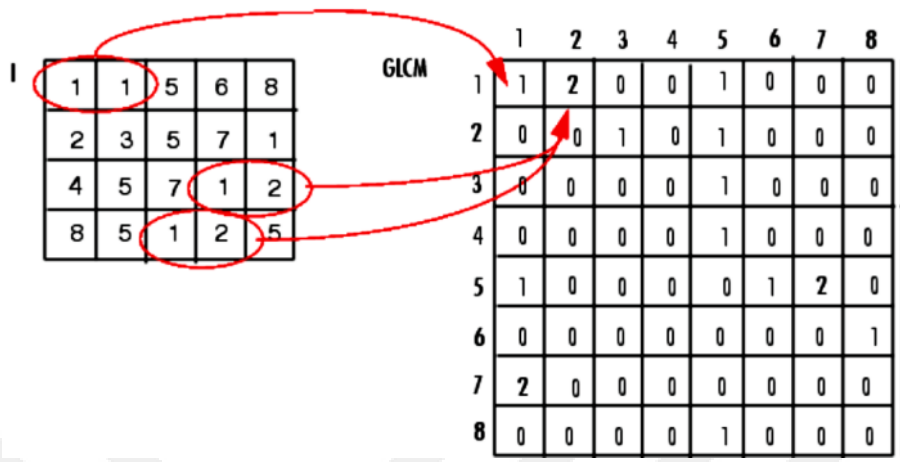

Figure 3.7: an example that shows how the GLCM is constructed

The GLCM can be used to extract a variety of statistical properties, having energy, entropy, variance, and more. Second-order statistics are used to calculate these characteristics [19]. Table 3.3 displays the various feature names and formulae.

Table 3.3: The different extracted features using GLCM for image analysis

No.	Feature Name	Equation
i.	-Energy	$f_1 = \sum_i \sum_j \{P(i,j)\}^2$
ii.	-Contrast	$f_2 = \sum_{n=0}^{N-1} n^2 \{\sum_{i=1}^{N} \sum_{j=1}^{N} P(i,j) \mid \lvert i-j \rvert = n\}$
iii.	-Correlation	$f_3 = \frac{\sum_i \sum_j (ij)P(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$
iv.	-Entropy	$f_4 = \sum_i \sum_j P(i,j) \log(p(i,j))$
v.	-Homogeneity	$f_5 = \sum_i \sum_j \frac{1}{1+(i-j)^2} P(i,j)$
vi.	-Autocorrelation	$f_6 = \sum_i \sum_j (ij)P(i,j)$

Continue …..		
No.	**Feature Name**	**Equation**
vii.	-Dissimilarity	$f_7 = \sum_i \sum_j \lvert i-j \rvert . P(i,j)$
viii.	-Cluster Shade	$f_8 = \sum_i \sum_j (i - j - \mu_x - \mu_y)^3 . P(i,j)$
ix.	-Cluster Prominence	$f_9 = \sum_i \sum_j (i + j - \mu_x - \mu_{y)}{}^4 . P(i,j)$
x.	-Maximum Probability	$f_{10} = \max_{i,j} p(i,j)$
xi.	-Sum Of Squares: Variance	$f_{11} = \sum_i \sum_j (i - \mu)^2 P(i,j)$
xii.	-Inverse Difference Moment	$f_{12} = \sum_i \sum_j \frac{1}{1 + (i - j)^2} P(i,j)$
xiii.	-Sum Variance	$f_{14} = \sum_{i=2}^{2N} (i - f_s)^2 P_{x+y}(i)$
xiv.	-Sum Entropy	$f_{15} = - \sum_{i=2}^{2N} P_{x+y}(i) \log\{P_{x+y}(i)\}$
xv.	-Difference Variance	f_{16} = Variance of p_{x-y}
xvi.	-Difference Entropy	$f_{17} = - \sum_{i=0}^{N-1} P_{x-y}(i) \log\{P_{x-y}(i)\}$

4. RESULT AND DISCUSSIONS

Images datasets were collected during this study as well as from the https://www.imaging-resource.com/IMCOMP/COMPS01.HTM data set of images taken by different cameras, Rainy images [90], and YTVOS201 [91] in clear and rainy weather conditions, images, or videos under normal conditions to obtain a high-resolution image of objects detection using the rain removal algorithm and evaluated using metrics of PSNR and SSIM, additional, the image test for object detection algorithm by (YOLOV3) before and after processing to confirm the accuracy of the developed methodology.

4.1 CONTRAST ENHANCEMENT

Here are some examples of results (low light image contrast enhancement with histogram) shown in Figures 4.1, 4.2, the first column indicates the input image, and the second image represents the output images.

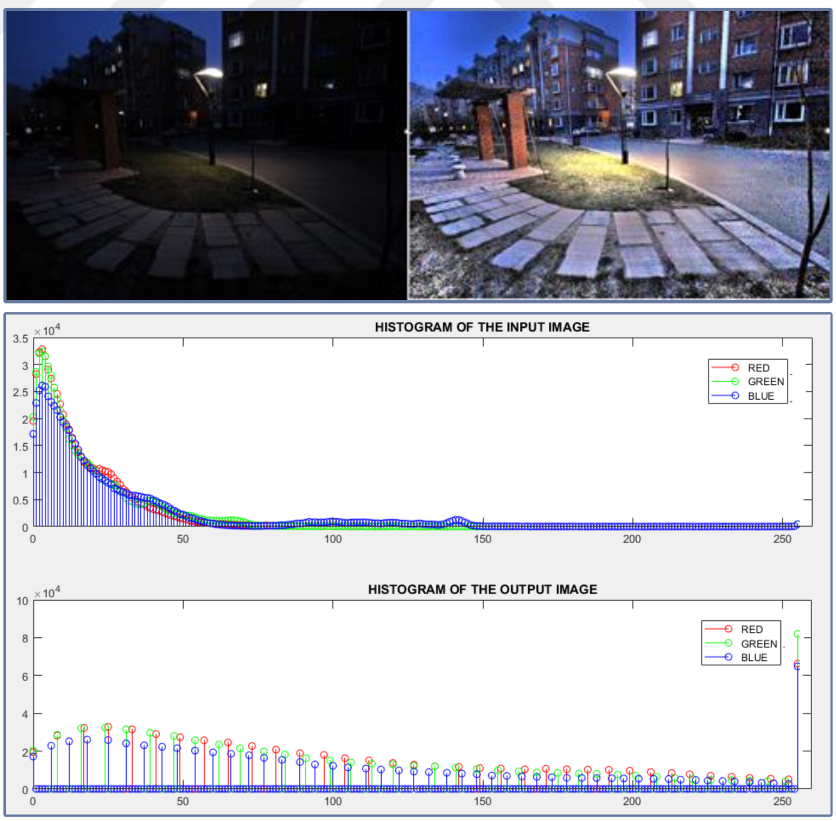

Figure 4.1.: Low contrast image enhancement 1

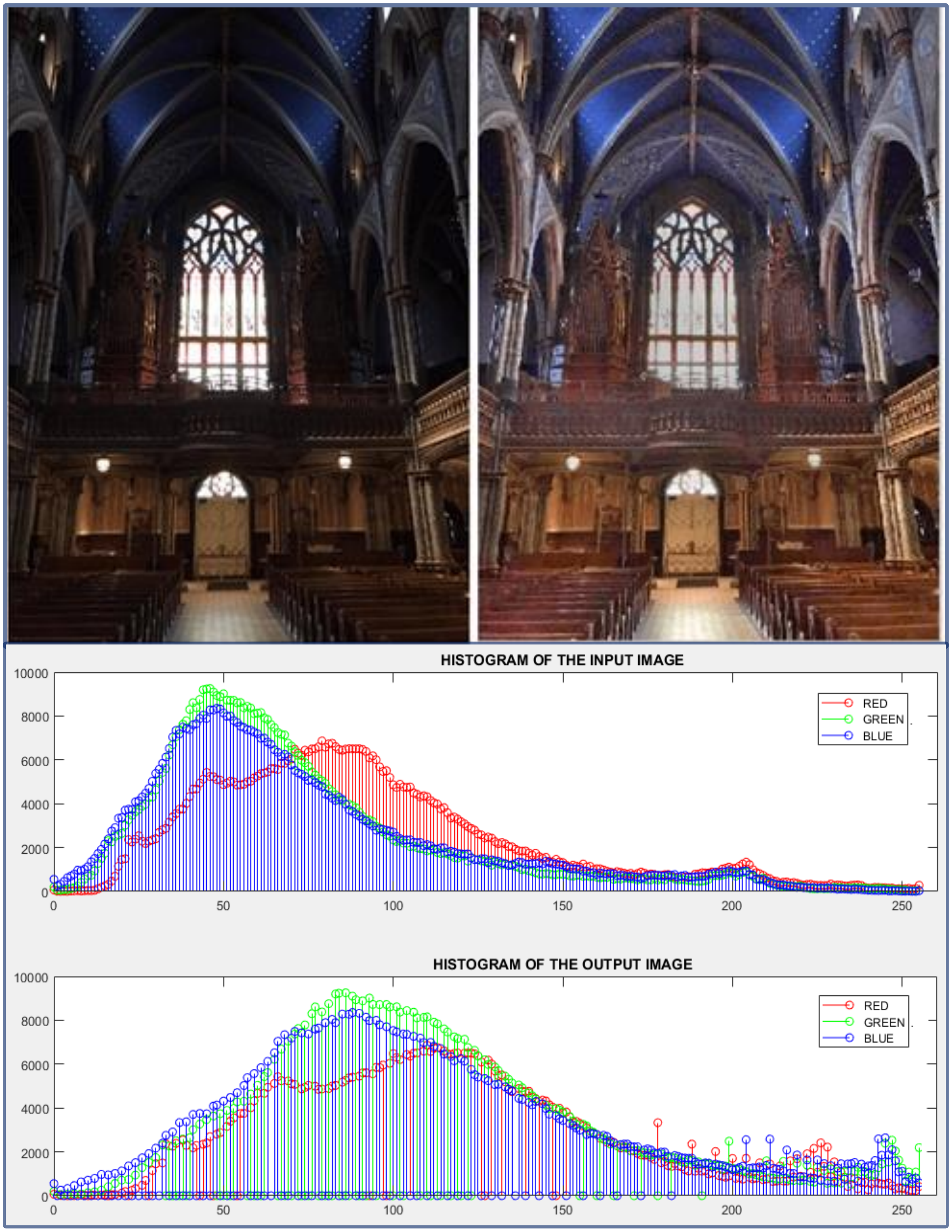

Figure 4.1: Low light image contrast enhancement 2

4.2 DE-RAINING RESULTS

Using two datasets [90][91], the experimental findings of the modified technique of de-raining picture and actual video de-raining are presented in Figure 4.3, and the dataset results are displayed in Figure 4.4.

Figure 4.3: Results of the de-raining algorithm on the YTVOS201 dataset

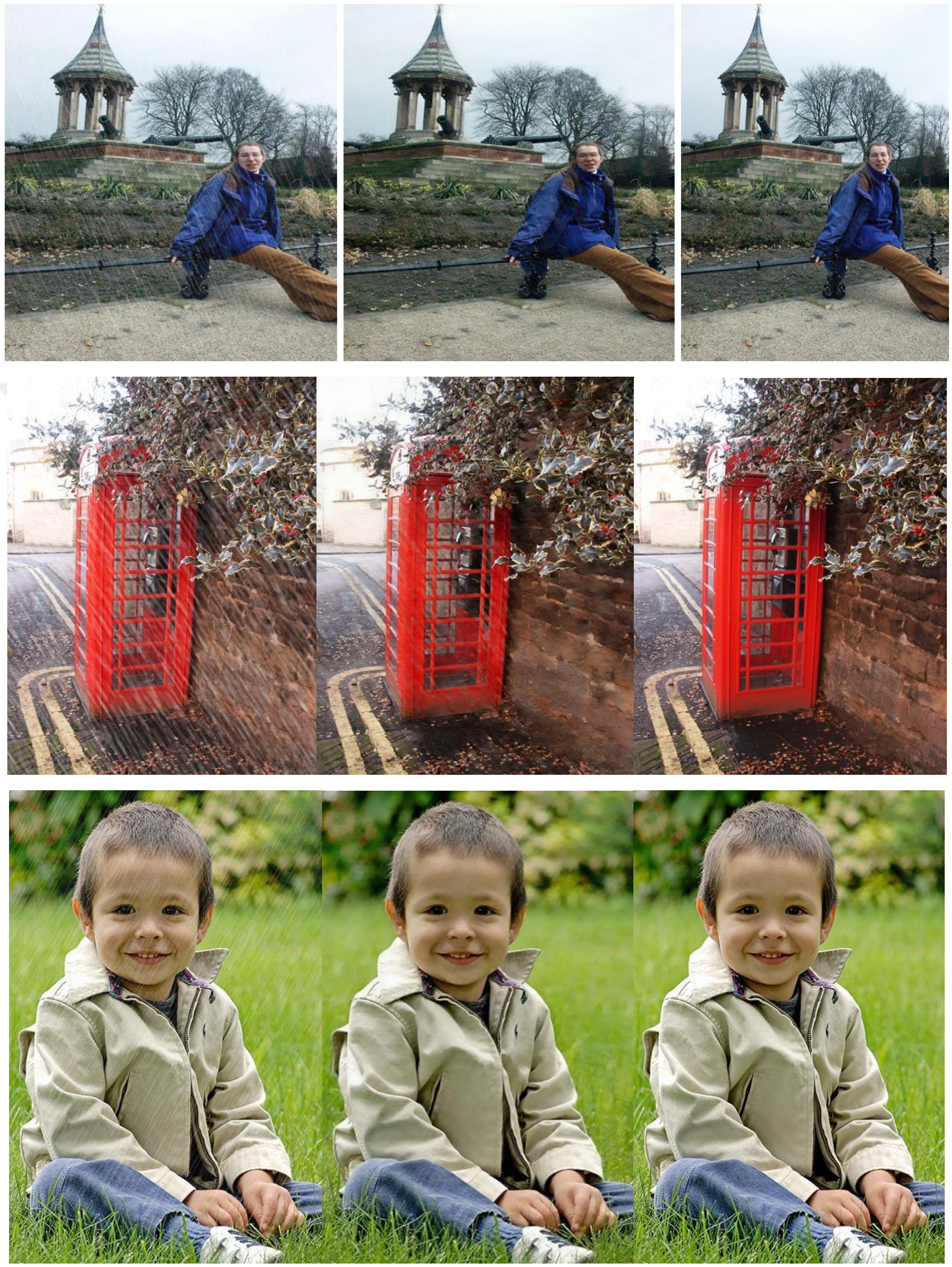

Figure 4.4: Results of the de-raining algorithm on a Rainy image dataset

On the synthetic pictures in Figures 4.3 and 4.4, Table 4.1 compares the quantitative findings of the several competing methodologies used to create those images. According to the table, we can infer that the rain removal method presented is more efficient.

Table 4.1: Evaluation of the performance of PSNR and SSIM on four synthetic images

Images	PSNR	SSIM	RAIN REMOVED %
Image 1	15.47	0.898	94 %
Image 2	17.33	0.915	95 %
Image 3	15.96	0.923	96 %
Image 4	18.51	0.982	97 %
Image 5	13.91	0.945	95 %
Image 6	14.62	0.952	98 %

The comparative findings with alternative rain removal technologies are shown in Figure 4.5 and Table 4.2.

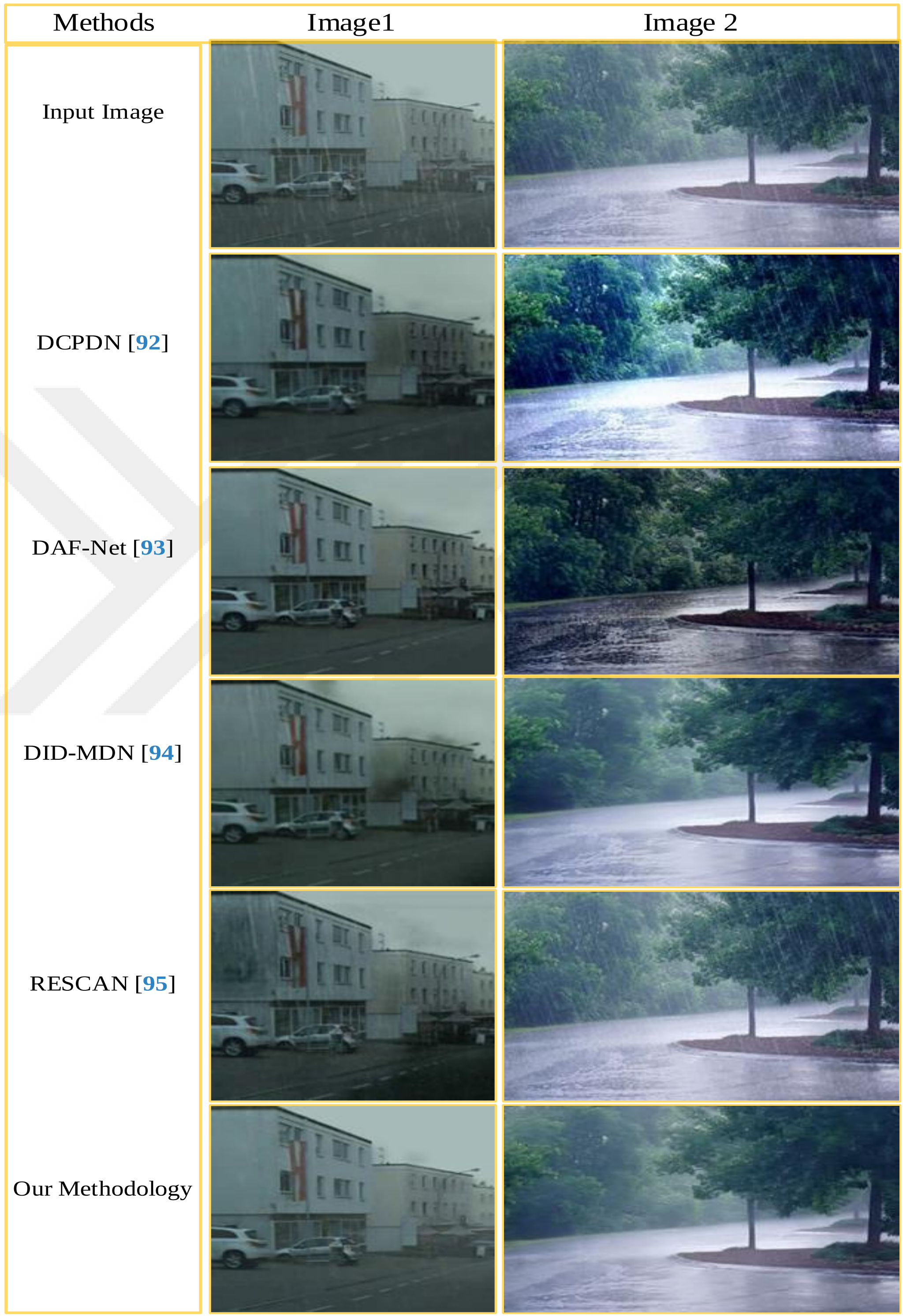

Figure 4.5: Results of rain removers using different methods with our method.

Table 4.2: Comparisons with other methods using PSNR and SSIM on the test Rainy image

Method	**PSNR**	**SSIM**
DCPDN [92]	18.588	0.8329
DAF-Net [93]	21.063	0.8713
DID-MDN [94]	18.863	0.8331
RESCAN [95]	18.238	0.8288
Our Method	31.374	0.9437

By comparing the results of the algorithm on both datasets, it's obvious that the model succeeds in removing the rain dashes from the rainy image dataset while some rain streaks are not removed completely or at all YTVOS201 datasets. The main reason could be that the rain streaks are bigger and not realistic in YTVOS201 dataset images, and this led to leaving some rain or sometimes shadows or marks in the images. From Table 4.2 and Figure 4.5, we can recognize that our results are clearer than other methods (the rate of error is less than all methods that are compared with), where PSNR is 31.374 and SSIM is 0.9437. The result of other methods collecting some of the rain and losing the image quality and the contrast of the image.

4.1.1 De-Raining With Object Detection Results

We have used two different datasets of images having different objects in size and location, the first dataset is Rainy image, which has 116 images (116 objects), and the second dataset is called YTVOS201, which has 13 images (33 objects). After we evaluated each section separately (object detection before and after de-raining). Data on the number of true positives, false positives, true negatives, and all four types of misclassification are summarized in Table 4.3.

Table 4.3: The Calculated Values of the Algorithms before and after De-raining

DATASET	**No of Objects**	Before processing			After processing		
		True Positives	**False Positives**	**False Negatives**	**True Positives**	**False Positives**	**False Negatives**
Rainy Images Dataset	**116**	90	11	15	**105**	**8**	**3**
YTVOS201 Dataset	**33**	20	10	3	**27**	**4**	**2**

From the values presented in Table 4.3, we get the results on each dataset presented in form of the Precision, Recall, and F1 scores. We can see that the average object detection results of our methodology (after processing) are more accurate than other methods such as before processing. Therefore, the results after processing are more accurate in detecting objects than before processing. The results on each dataset are presented in form of the Precision, Recall, and F1 scores as in Figure 4.6.

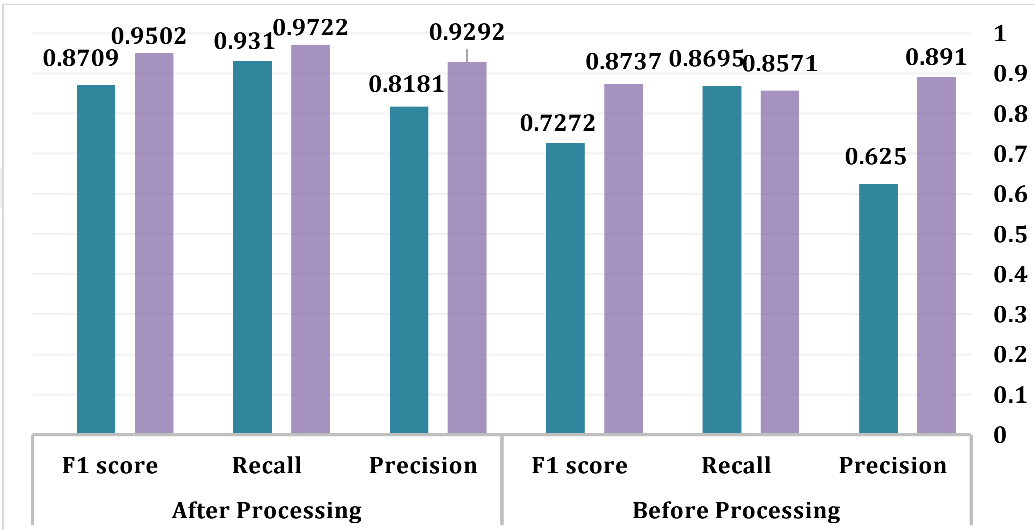

Figure 4.6: Performance evaluation of object detection before and after processing images

From Figure 4.6, we can recognize that the object detection of better than before processing, and as much as the results of our methodology are more accurate than others, which indicates the proposed methodology is more useful to use.

Rain removal and contrast enhancement are more effective at recognizing tiny things or objects in challenging settings, such as objects showing partially in a frame, due to the technology's ability to execute object detection process at three distinct scales. The items look smaller because they are positioned on the far side of the scaled location.

4.1.2 Qualitative Comparison of the Proposed Method

The Rainy Images dataset is more suitable for object detection than YTVOS201dataset in terms of the variety of objects that exist in images, but the synthesized rain streaks are extremely and abnormally big. Therefore, the result of the de-raining algorithm is not successful as the results of the rainy image dataset or even as real rain images as shown in Figure 4.6. As we can see from images the residual rain streaks in images lead to disorder in the borders of the object or even blur some objects, eventually the object detection and identification algorithms fails to detect and recognize some objects. Also, this dataset was constructed by extracting frames from YouTube videos, because of motion these images (frames) are not always clear which affects the enhancement and detection accuracy.

The Rainy image dataset although has a very good performance in quality enhancement, the general concept of the dataset is not about object detection but since we could not find any joint dataset for rainy image and object detection, we choose this one. In Figure 4.7 some examples of motion effects on images and objects.

(a) Rainy images (b) Rain removed

Figure 4.7: Examples from datasets [90], [91] show the movement of false detected objects after the rain removal algorithm applied to the images.

Which are very promising results for the enhancement algorithm. Although the true detection rates are not very satisfying the improvement in the true and false detection rates before and after rain is remarkable.

Eventually, the enhancement (de-raining) algorithm’s most important property is removing uncertainty from detection, which means quite a high number of false detected or false classified objects after the rain removal algorithm become true detected or classified as seen in Figure 4.7. Even though it is not affected or seen in Table 4.3, there was a high number of objects were detected with high detection rates (more certainty). This means the de-raining algorithm not only removed uncertainty from false detections but also emphasized or detected and classified the true objects with higher confidence as shown in Figure 4.8, the true detection on the rainy image is 97.33 and on removed rain is 99.92.

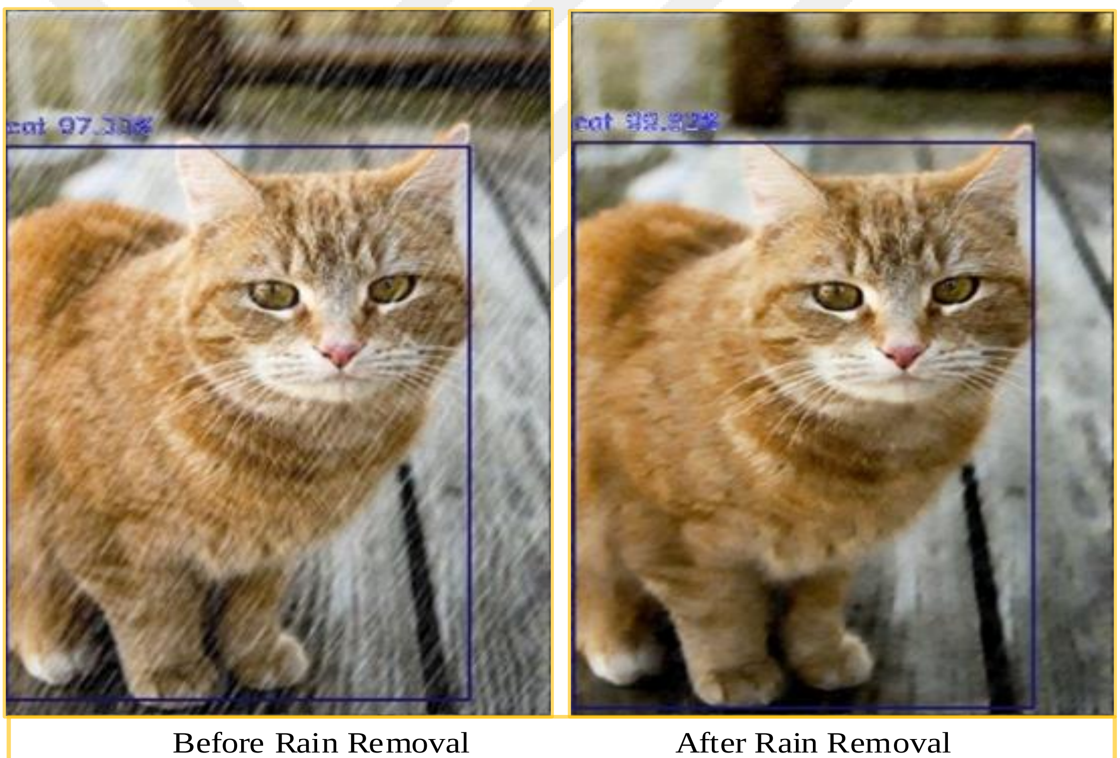

Figure 4.8: The image sample from the rainy image dataset shows the improvement in the detection rate after the rain removal algorithm was applied.

In Figure 4.9 and Figure 4.10, we have applied our methodology (De-raining with YOLOV3) on real rainy images, which compeered with other modern methods and with different rain level (Light rain, Medium rain, and Heavy rain)

Figure 4.9: Qualitative comparison of the proposed algorithm on the synthetic datasets.

Input Images **Pedestrian detection mode [96]** **Our (De-raining+YOLOV3)**

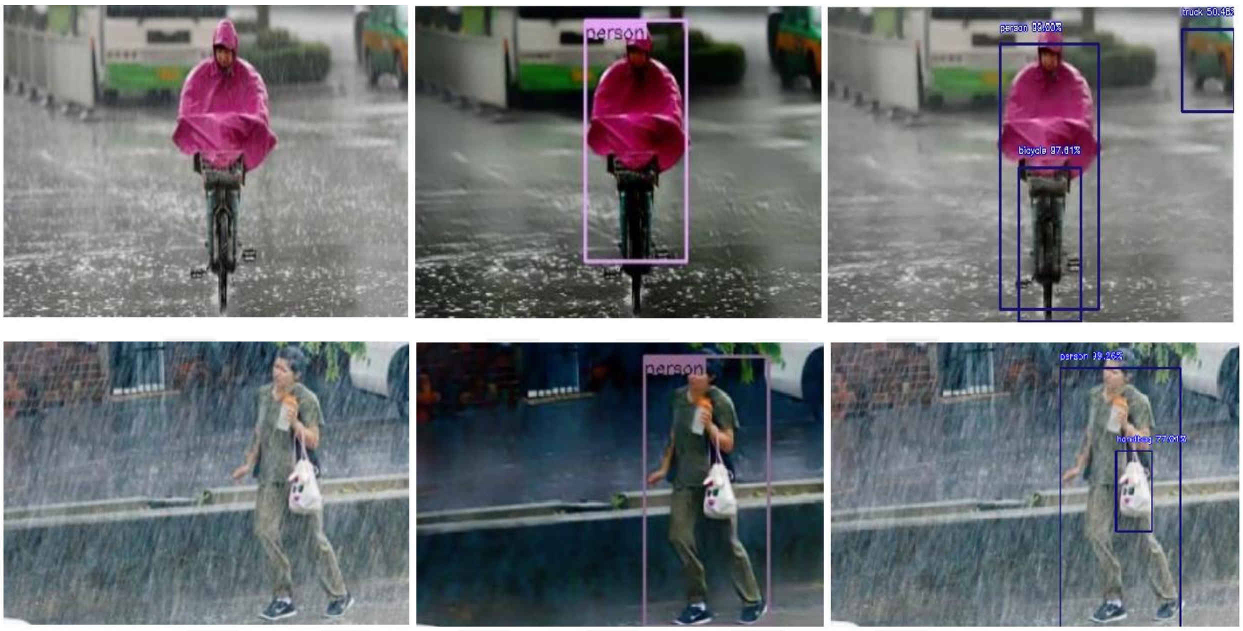

Figure 4.10: Qualitative comparison of the proposed algorithm on real-world rainy images.

Figures 4.9 and 4.10 demonstrate that the proposed methodology recovers the original image's identification effectiveness. The presented approach improves average object identification in light, medium, and severe rain conditions. The findings further demonstrate the importance of the de-raining module. Even after testing pedestrian datasets on rainfall events, previous systems were unable to properly recognize pedestrians. On rainy days, on the other hand, our approach can correctly identify pedestrians.

The accuracy of the models for the predicated objects has been calculated to be 87.06 % for YOLOv3 before applying our de-raining method and 96.55 % after implementing our method (De-raining) with YOLOv3 on rainy datasets, while on the YTVOS201 Dataset the accuracy before processing images has been calculated to be 90.90% for YOLOv3 and 93.96% after processing images with De-raining and YOLOv3.

4.2 IMAGE FEATURES ANALYSIS WITH DIFFERENT CAMERAS

After choosing the scene, 16 digital cameras are used to take pictures, with and without the flashlight, thus 16 pictures were created. It is worth noting that all these cameras differ in resolution. Table 4.4 shows the camera types and their resolutions.

Table 4.4: Device type of taking pictures with their resolution

Device name	Resolution with image zoom
Canon 5D Mark IV	30.40 Megapixels ~ Zoom: n/a (Pro SLR)
Canon EOS 6D Mark II	26.20 Megapixels ~ Zoom: 4.38x
Canon EOS M50	24.20 Megapixels ~ Zoom: 3.00x
Canon Power-Shot G9 X Mark II	20.20 Megapixels ~ Zoom: 3.00x
Fujifilm X-A5	24.20 Megapixels ~ Zoom: 3.00x
Nikon D850	45.70 Megapixels ~ Zoom: n/a (Pro SLR)
Nikon D3500	24.20 Megapixels ~ Zoom: 3.06x
Olympus OM-D E-M10 II	16.10 Megapixels ~ Zoom: 3.00x
Panasonic Lumix DC-GH5	20.30 Megapixels ~ Zoom: n/a (Non-Zoom)
Panasonic Lumix DMC-LX10	20.10 Megapixels ~ Zoom: 3.00x
Pentax 645Z	51.40 Megapixels ~ Zoom: n/a (Pro SLR)
Pentax K-70	24.24 Megapixels ~ Zoom: 7.50x
Ricoh GR II	16.20 Megapixels ~ Zoom: 1.00x
Sony Alpha ILCE-A7	24.30 Megapixels ~ Zoom: 2.50x
Sony Cyber-shot DSC-RX10	20.20 Megapixels ~ Zoom: 8.33x
Sony Alpha ILCE-A7R II	42.40 Megapixels ~ Zoom: n/a (Non-Zoom)

Figures 4.11 and 4.12 show the images taken by different cameras before and after rain removal.

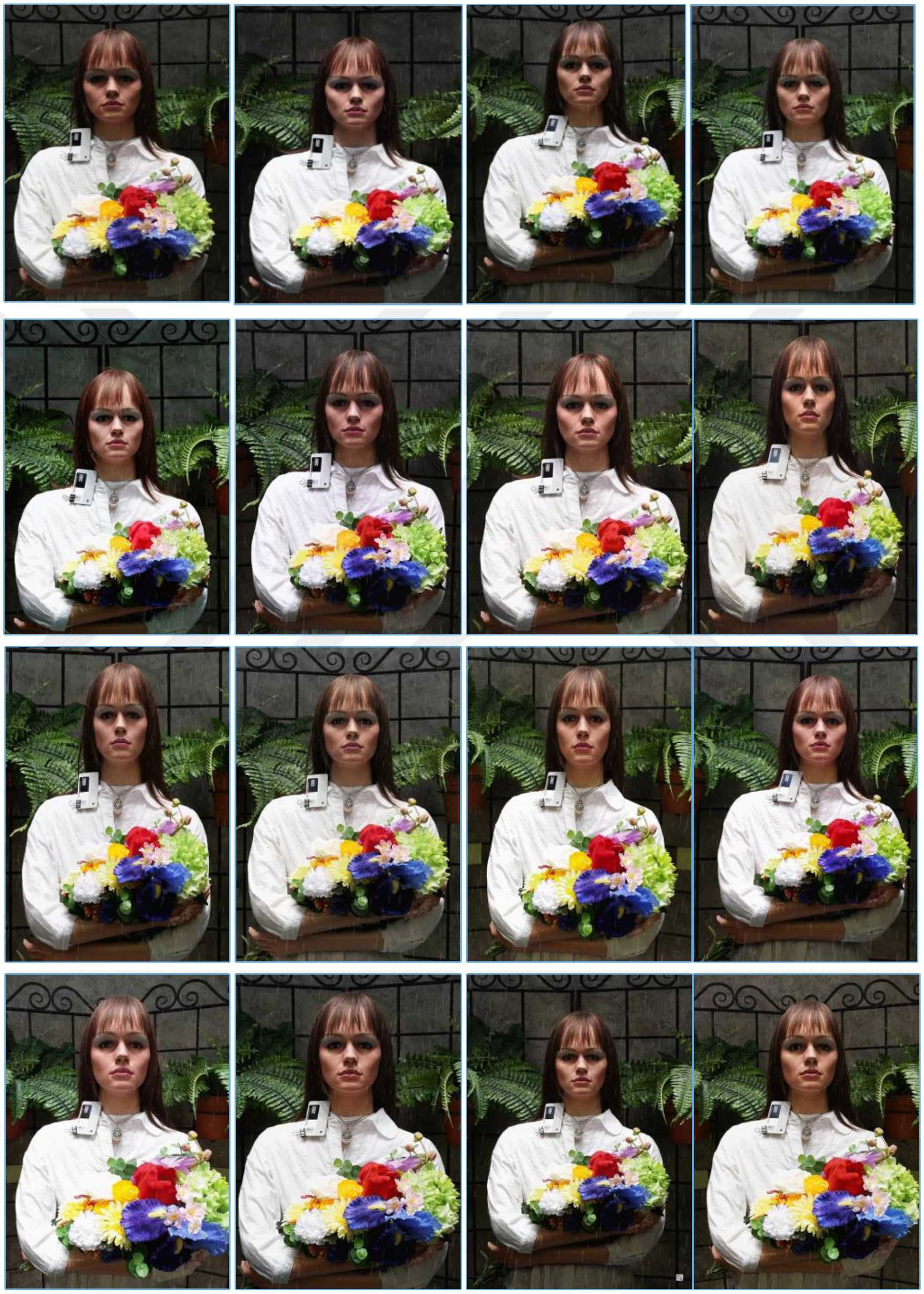

Figure 4.11: Image with rain taken under different cameras

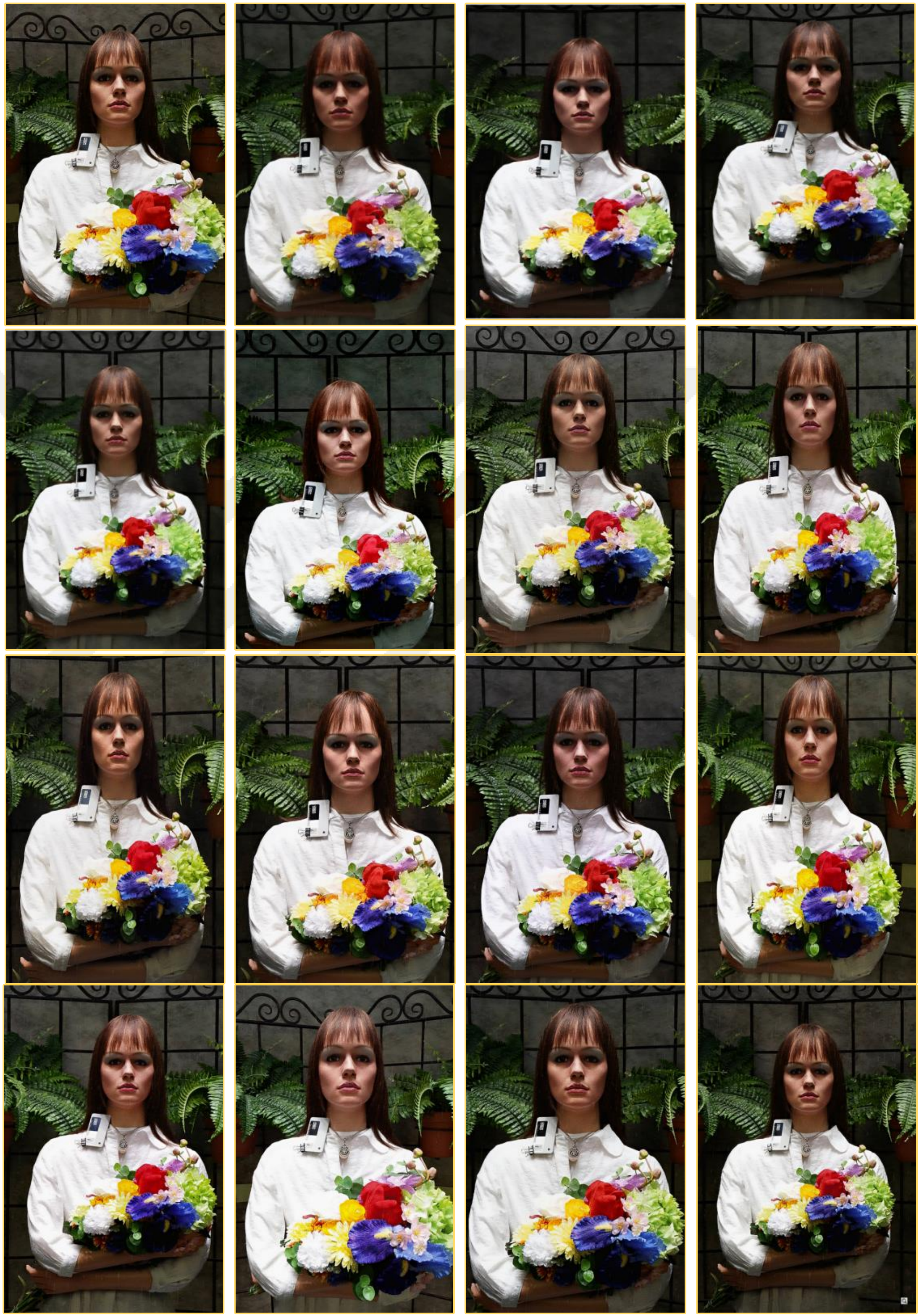

Figure 4.12: Rain removal images taken under different cameras

4.2.1 Leveling

After gathering the results, the numbers (feature values) should be substituted by levels of Low, Medium, and High to differentiate them easily by letters, not by pure numbers. The next formulas describe intervals are dependent to convert numbers to levels, where: A represents the Minimum value of the feature, B represents the Maximum value, division by 3 stands for the 3 levels required. The results after leveling are shown in Tables 4.5 and 4.6.

Low Range:

$$mini\ low = A\ , max\ low = (B - A)/3 + A \tag{4.1}$$

Medium Range:

$$mini\ medium = (B - A)/3 + A, \tag{4.2}$$

which is the same as the max low.

$$max\ medium = B - (B - A)/3 \tag{4.3}$$

High Range:

$$mini\ high = B - (B - A)/3, \tag{4.4}$$

which is the same as the max medium. $max\ high = B$.

Table 4.5: Leveling the results into Low, Medium, and High Values of rainy images

Device name	Contrast	Correlation	Energy	Homogeneity	Mean	Standard Deviation	Entropy	RMS	Variance	Smoothness	Kurtosis	Skewness	IDM
Canon 5D Mark IV	ML	ML	ML	ML	LL	HL	ML	HL	HL	HL	ML	LL	LL
Canon EOS 6D Mark II	HL	LL	LL	LL	HL	HL	ML	HL	HL	HL	HL	HL	ML
Canon EOS M50	ML	HL	ML	ML	LL	HL	LL	HL	HL	HL	HL	HL	HL
Canon Power-Shot G9 X Mark II	ML	ML	ML	ML	LL	HL	ML	HL	HL	LL	ML	ML	ML
Fujifilm X-A5	LL	LL	ML	ML	HL	HL	HL	HL	ML	HL	LL	LL	ML
Nikon D850	LL	HL	ML	ML	HL	HL	HL	HL	ML	HL	ML	ML	HL
Nikon D3500	LL	LL	ML	ML	HL	HL	ML	HL	ML	HL	LL	ML	ML
Olympus OM-D E-M10 II	LL	HL	HL	HL	HL	LL	LL	LL	LL	HL	HL	HL	ML
Panasonic Lumix DC-GH5	LL	HL	HL	HL	HL	LL	LL	LL	LL	HL	LL	ML	ML
Panasonic Lumix DMC-LX10	ML	ML	ML	ML	HL	HL	HL	HL	ML	HL	ML	HL	HL
Pentax 645Z	LL	LL	HL	HL	ML	LL	LL	LL	LL	HL	HL	HL	LL
Pentax K-70	ML	ML	ML	ML	ML	HL	ML	HL	HL	HL	HL	HL	HL
Ricoh GR II	ML	ML	ML	ML	HL	HL	HL	HL	ML	HL	LL	HL	ML
Sony Alpha ILCE-A7	ML	ML	ML	ML	HL	HL	HL	HL	ML	HL	LL	HL	ML
Sony Cyber-shot DSC-RX10	LL	LL	HL	HL	LL	HL	ML	HL	ML	HL	ML	ML	HL
Sony Alpha ILCE-A7R II	ML	HL	ML	ML	HL	HL	ML	HL	ML	HL	LL	ML	HL

Table 4.6: Leveling the results into Low, Medium, and High Values of rain removal

Device name	Contrast	Correlation	Energy	Homogeneity	Mean	StandardDeviation	Entropy	RMS	Variance	Smoothness	Kurtosis	Skewness	IDM
Canon 5D Mark IV	ML	LL	ML	ML	LL	HL	LL	HL	HL	LL	HL	LL	LL
Canon EOS 6D Mark II	HL	LL	LL	LL	ML	HL	HL	HL	HL	LL	HL	ML	ML
Canon EOS M50	HL	HL	LL	LL	ML	HL	ML	HL	HL	LL	HL	ML	ML
Canon Power-Shot G9 X Mark II	ML	HL	ML	ML	LL	HL	ML	HL	HL	HL	ML	ML	ML
Fujifilm X-A5	ML	LL	LL	LL	HL	HL	HL	HL	ML	LL	LL	ML	HL
Nikon D850	LL	ML	ML	HL	ML	HL	ML	HL	ML	LL	ML	LL	ML
Nikon D3500	LL	LL	ML	ML	ML	HL	HL	HL	ML	LL	LL	LL	LL
Olympus OM-D E-M10 II	LL	LL	HL	HL	ML	LL	LL	LL	LL	LL	HL	ML	HL
Panasonic Lumix DC-GH5	LL	LL	HL	HL	ML	LL	LL	LL	LL	LL	ML	ML	ML
Panasonic Lumix DMC-LX10	ML	ML	ML	ML	HL	HL	HL	HL	ML	LL	LL	ML	ML
Pentax 645Z	LL	LL	HL	HL	ML	LL	LL	LL	LL	LL	HL	ML	ML
Pentax K-70	ML	ML	ML	ML	ML	HL	ML	HL	HL	LL	ML	HL	ML
Ricoh GR II	HL	LL	LL	LL	HL	HL	HL	HL	ML	LL	ML	HL	ML
Sony Alpha ILCE-A7	ML	HL	ML	ML	HL	HL	HL	HL	ML	LL	LL	ML	HL
Sony Cyber-shot DSC-RX10	ML	LL	ML	ML	ML	HL	ML	HL	ML	LL	ML	HL	ML
Sony Alpha ILCE-A7R II	ML	LL	LL	LL	HL	HL	HL	HL	ML	LL	LL	HL	ML

Depending on Tables 4.5 and 4.6, it is easy to classify the features in the Low and High camera resolutions in the presence. As we have the level of the camera resolution from the Leveling part described previously, we can recognize that more features decreased to be Low rather than it was High or medium.

To focus more on each feature individually according to the same camera type (resolution), with dual cases of the processing of rain removal, Table 4.7 of all changed features are created to compare results in both Histogram and Texture features before and after processing of rain remove on 16 different pictures taken by different cameras resolution.

Table 4.7: Feature analysis before and after image rain removal

	Before Processing images with different cameras (without De-raining)												
Features Level	Smoothness	Correlation	IDM	Skewness	Kurtosis	Energy	Homogeneity	Mean	Standard Deviation	RMS	Variance	Contrast	Entropy
Low	-0.3809	0.0461	-1.9382	0.1664	5.0846	0.8558	0.9620	0.0000	0.0611	0.0611	0.0037	0.0660	3.3173
Max low & Mini Medium	0.9688	0.0762	1.5105	0.2893	5.7491	0.8944	0.9728	0.0015	0.0662	0.0662	0.0044	0.0958	3.4237
Max Medium & Mini Max	0.0690	0.0562	-0.7886	0.2074	5.3061	0.8687	0.9656	0.0005	0.0628	0.0628	0.0039	0.0759	3.3527
Max	0.5189	0.0662	0.3609	0.2484	5.5276	0.8816	0.9692	0.0010	0.0645	0.0645	0.0042	0.0859	3.3882
	After Processing images with different cameras (with De-raining)												
	Increased Features							Not changed Features				Decreased Features	
Low	0.8892	0.0490	-1.683382	0.1695	5.128573	0.8601	0.9632	-0.0004	0.0611	0.0611	0.0037	0.0665	3.3073
Max low & Mini Medium	2.8499	0.0900	1.928876	0.3359	5.848798	0.8953	0.9729	0.0017	0.0662	0.0662	0.0044	0.0911	3.4083
Max Medium & Mini Max	1.5427	0.0626	-0.479296	0.2250	5.368648	0.8718	0.9664	0.0003	0.0628	0.0628	0.0039	0.0747	3.3410
Max	2.1963	0.0763	0.72479	0.2804	5.608723	0.8835	0.9697	0.0010	0.0645	0.0645	0.0042	0.0829	3.3746

5. CONCLUSIONS AND FUTURE WORK

5.1 CONCLUSION

The proposed methodology for image enhancement and features analysis based on deep learning and rain removal in images to increase the accuracy of object detection and reduce the rate of false under rain conditions is discussed. According to the findings, the combined methodology (contrast enhancement and De-raining) showed the best performance to detect objects and suit the safety camera application. This work mainly deals with detecting objects in rain conditions circumstances. The types are limited to a great extent. It was shown that this network may be used for real-world photos by synthesizing clean/rainy image pairings for learning. We demonstrated that deep learning using convolutional neural networks, a commonly used method for high-level vision tasks, can also be utilized to effectively cope with real photos in adverse weather. With regard to picture quality and computational efficiency, we found that Derain-Net significantly surpasses current approaches. In addition, we were able to demonstrate that we don't require a particularly deep (or broad) network to do this job by employing image processing domain knowledge.

5.2 FUTURE WORK

Rain removal from still photographs is more difficult than from moving video because the pixels of the scene is obscured or blurred when the images are acquired. This is why we have concentrated our study on single-image rain removal. Without this information, it is possible to remove rain pixels without deleting all of the rain pixels in a video, resulting in a loss of certain rain pixels and blurring of scene material in some complicated cases. In order to fully use the Spatio-temporal information in the video, we propose that this practical technique be extended beyond the removal of rain from a single picture to the removal of rain from the video. Recurrent Neural Networks (RNNs) and CycleGANs may be coupled to build a Recurrent CycleGAN network, which can be used to totally eliminate rain from video applications. Developing our mode to be used in several important applications in our daily lives, such as security cameras, self-driving cars, and other obstacle detection systems. Connect the system to a weather application where we can know if it's raining or not if it is raining activate the de-raining algorithm before object recognition.

REFERENCES

[1] W. Yang, R. T. Tan, J. Feng, Z. Guo, S. Yan, and J. Liu, “Joint rain detection and removal from a single image with contextualized deep networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1377–1393, 2020.

[2] K. Park, M. Kim, H. Lim, S. Yu, and J. Paik, “Fast rain removal using rain map and temporal filtering for consumer imaging applications,” in *IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, 2018.

[3] C. H. Bahnsen and T. B. Moeslund, “Rain Removal in Traffic Surveillance: Does it Matter?,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 2802–2819, 2019.

[4] J. Xu, W. Zhao, P. Liu, and X. Tang, “Removing rain and snow in a single image using guided filter,” in *IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, 2012.

[5] Jie Chen and Lap-Pui Chau, “A rain pixel recovery algorithm for videos with highly dynamic scenes,” *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1097–1104, 2014.

[6] D.-Y. Chen, C.-C. Chen, and L.-W. Kang, “Visual depth guided color image rain streaks removal using sparse coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 8, pp. 1430–1455, 2014.

[7] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, “Deep joint rain detection and removal from a single image,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[8] H. Wang, Q. Xie, Q. Zhao, and D. Meng, “A model-driven deep neural network for single image rain removal,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[9] A. Mohammed, “Analysis of image features taken from different camera types,” *Muthanna Journal of Engineering and Technology*, vol. 6, no. 2, pp. 137–151, 2018.

[10] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, “Deep joint rain detection and removal from a single image,” *arXiv [cs.CV]*, 2016.

[11] U. Sara, M. Akter, and M. S. Uddin, “Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study,” *J. Comput. Commun.*, vol. 07, no. 03, pp. 8–18, 2019.

[12] Y.-F. Liu, D.-W. Jaw, S.-C. Huang, and J.-N. Hwang, “DesnowNet: Context-aware deep network for snow removal,” *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 3064–3073, 2018.

[13] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, “Automatic single-image-based rain streaks removal via image decomposition,” *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1742–1755, 2012.

[14] T. Han, C. Liu, W. Yang, and D. Jiang, “Learning transferable features in deep convolutional neural networks for diagnosing unseen machine conditions,” *ISA Trans.*, vol. 93, pp. 341–353, 2019.

[15] J.-H. Kim, C. Lee, J.-Y. Sim, and C.-S. Kim, “Single-image deraining using an adaptive nonlocal means filter,” in *IEEE International Conference on Image Processing*, 2013.

[16] D. Eigen, D. Krishnan, and R. Fergus, “Restoring an image taken through a window covered with dirt or rain,” in *IEEE International Conference on Computer Vision*, 2013.

[17] Y.-L. Chen and C.-T. Hsu, “A generalized low-rank appearance model for spatio-temporally correlated rain streaks,” in *IEEE International Conference on Computer Vision*, 2013.

[18] D.-A. Huang, L.-W. Kang, M.-C. Yang, C.-W. Lin, and Y.-C. F. Wang, “Context-aware single image rain removal,” in *IEEE International Conference on Multimedia and Expo*, 2012.

[19] Y. Wang, Y. Song, C. Ma, and B. Zeng, “Rethinking image deraining via rain streaks and vapors,” *arXiv [cs.CV]*, 2020.

[20] T. Matsui, T. Fujisawa, T. Yamaguchi, and M. Ikehara, “Single-image rain removal using residual deep learning,” in *IEEE International Conference on Image Processing (ICIP)*, 2018.

[21] W. Yang, S. Wang, D. Xu, X. Wang, and J. Liu, “Towards scale-free rain streak removal via self-supervised fractal band learning,” *Proc. Conf. AAAI Artif. Intell.*, vol. 34, no. 07, pp. 12629–12636, 2020.

[22] Q. Zhu, J. Yuan, and L. Shao, “The current challenges and prospects of rain detection and removal from videos,” in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2015.

[23] Y. Wang, Y. Song, C. Ma, and B. Zeng, “Rethinking image deraining via rain streaks and vapors,” in *Computer Vision – ECCV*, Cham: Springer International Publishing, 2020, pp. 367–382.

[24] Q. Wu, L. Wang, K. N. Ngan, H. Li, and F. Meng, “Beyond synthetic data: A blind deraining quality assessment metric towards authentic rain image,” in *IEEE International Conference on Image Processing (ICIP)*, 2019.

[25] Y. Luo, Y. Xu, and H. Ji, “Removing rain from a single image via discriminative sparse coding,” in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[26] C.-Y. Chen, C.-W. Hsieh, P.-W. Chi, C.-F. Lin, C.-J. Weng, and C.-H. Hwang, “High-speed image velocimetry system for rainfall measurement,” *IEEE Access*, vol. 6, pp. 20929–20936, 2018.

[27] X. Ding, L. Chen, X. Zheng, Y. Huang, and D. Zeng, “Single image rain and snow removal via guided L0 smoothing filter,” *Multimed. Tools Appl.*, vol. 75, no. 5, pp. 2697–2712, 2016.

[28] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, “Rain streak removal using layer priors,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[29] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and J. Paisley, “Clearing the skies: A deep network architecture for single-image rain removal,” *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2944–2956, 2017.

[30] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the Skies: A deep network architecture for single-image rain removal," *arXiv [cs.CV]*, 2016.

[31] L.-J. Deng, T.-Z. Huang, X.-L. Zhao, and T.-X. Jiang, "A directional global sparse model for single image rain removal," *Appl. Math. Model.*, vol. 59, pp. 662–679, 2018.

[32] J. Luka, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE trans. inf. forensics secur.*, vol. 1, no. 2, pp. 205–214, 2006.

[33] Z. Tang, M. Yu, H. Yao, H. Zhang, C. Yu, and X. Zhang, "Robust image hashing with singular values of quaternion SVD," *Comput. J.*, vol. 64, no. 11, pp. 1656–1671, 2021.

[34] J. Korhonen and J. You, "Peak signal-to-noise ratio revisited: Is simple beautiful?," in *Fourth International Workshop on Quality of Multimedia Experience*, 2012.

[35] T. Baar, W. van Houten, and Z. Geradts, "Camera identification by grouping images from database, based on shared noise patterns," *arXiv [cs.CV]*, 2012.

[36] G. J. Bloy, "Blind camera fingerprinting and image clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 532–535, 2008.

[37] B. Gupta and M. Tiwari, "Improving source camera identification performance using DCT based image frequency components dependent sensor pattern noise extraction method," *Digit. investig.*, vol. 24, pp. 121–127, 2018.

[38] J. Bernacki, "Digital camera identification based on analysis of optical defects," *Multimed. Tools Appl.*, vol. 79, no. 3–4, pp. 2945–2963, 2020.

[39] A. Rashidi and F. Razzazi, "Single image camera identification using I-vectors," in *7th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2017.

[40] M. Goljan, M. Chen, P. Comesaña, and J. Fridrich, "Effect of compression on sensor-fingerprint based camera identification," *IS&T Int. Symp. Electron. Imaging*, vol. 28, no. 8, pp. 1–10, 2016.

[41] R. G. VidalMata *et al.*, "Bridging the gap between computational photography and visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4272–4290, 2021.

[42] F. L. Siena, B. Byrom, P. Watts, and P. Breedon, "Utilising the Intel RealSense camera for measuring health outcomes in clinical research," *J. Med. Syst.*, vol. 42, no. 3, p. 53, 2018.

[43] S. Widiyanto, Y. Sukra, S. Madenda, D. T. Wardani, and E. P. Wibowo, "Texture feature extraction based on GLCM and DWT for beef tenderness classification," in *Third International Conference on Informatics and Computing (ICIC)*, 2018.

[44] B. Leng, S. Guo, X. Zhang, and Z. Xiong, "3D object retrieval with stacked local convolutional autoencoder," *Signal Processing*, vol. 112, pp. 119–128, 2015.

[45] J. Bossu, N. Hautière, and J.-P. Tarel, "Rain or snow detection in image sequences through use of a histogram of orientation of streaks," *Int. J. Comput. Vis.*, vol. 93, no. 3, pp. 348–367, 2011.

[46] D. Huiying and Z. Xuejing, "Detection and removal of rain and snow from videos based on frame difference method," in *The 27th Chinese Control and Decision Conference (CCDC)*, 2015.

[47] A. K. Tripathi and S. Mukhopadhyay, "Meteorological approach for detection and removal of rain from videos," *IET Comput. Vis.*, vol. 7, no. 1, pp. 36–47, 2013.

[48] X. Xue, X. Jin, C. Zhang, and S. Goto, “Motion robust rain detection and removal from videos,” in *IEEE 14th International Workshop on Multimedia Signal Processing (MMSP)*, 2012.

[49] N. Brewer and N. Liu, “Using the shape characteristics of rain to identify and remove rain from video,” in *Lecture Notes in Computer Science*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 451–458.

[50] P. C. Barnum, S. Narasimhan, and T. Kanade, “Analysis of rain and snow in frequency space,” *Int. J. Comput. Vis.*, vol. 86, no. 2–3, pp. 256–274, 2010.

[51] X. Zhang, H. Li, Y. Qi, W. Leow, and T. Ng, “Rain removal in video by combining temporal and chromatic properties,” in *IEEE International Conference on Multimedia and Expo*, 2006.

[52] D.-A. Huang, L.-W. Kang, Y.-C. F. Wang, and C.-W. Lin, “Self-learning based image decomposition with applications to single image denoising,” *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 83–93, 2014.

[53] S.-H. Sun, S.-P. Fan, and Y.-C. F. Wang, “Exploiting image structural similarity for single image rain removal,” in *IEEE International Conference on Image Processing (ICIP)*, 2014.

[54] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, “Removing rain from single images via a deep detail network,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[55] U. Güçlü and M. A. J. van Gerven, “Deep neural networks reveal a gradient in the complexity of neural representations across the brain’s ventral visual pathway,” *arXiv [q-bio.NC]*, 2014.

[56] N. Krüger *et al.*, “Deep hierarchies in the primate visual cortex: what can we learn for computer vision?,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1847–1871, 2013.

[57] B. Hansen *et al.*, “SparseMeshCNN with self-attention for segmentation of large meshes,” *nldl*, vol. 3, 2022.

[58] M. Marti, S. Bujwid, A. Pieropan, H. Azizpour, and A. Maki, “An analysis of over-sampling labeled data in semi-supervised learning with FixMatch,” *nldl*, vol. 3, 2022.

[59] S. Das, N. L. Lund, N. N. Lønfeldt, A. K. Pagsberg, and L. H. Clemmensen, “Continuous metric learning for transferable speech emotion recognition and embedding across low-resource languages,” *nldl*, vol. 3, 2022.

[60] M. Sarmad, L. Ruspini, and F. Lindseth, “Photo-realistic continuous image super-resolution with implicit neural networks and generative adversarial networks,” *nldl*, vol. 3, 2022.

[61] Y. Xing, Deep Learning and Neuromorphic Lab, Centre for Image and Signal Processing, Electronics and Electrical Engineering Department University of Strathclyde, Glasgow, UK, W. Ke, G. D. Caterina, and J. Soraghan, “Noise reduction using neural lateral inhibition for speech enhancement,” *Int. J. Mach. Learn. Comput.*, vol. 11, no. 5, pp. 357–361, 2021.

[62] K. Upadhyay, M. Agrawal, and D. Deepak, “Ensemble learning-based COVID-19 detection by feature boosting in chest X-ray images,” *IET Image Process.*, vol. 14, no. 16, pp. 4059–4066, 2020.

[63] J. Kun, Z. Zhenhai, Y. Jiale, and D. Jianwu, “A deep learning-based method for pixel-level crack detection on concrete bridges,” *IET Image Process.*, 2022.

[64] K. M. T. Pöhlmann, L. O’Hare, P. Dickinson, A. Parke, and J. Föcker, “Action video game players do not differ in the perception of contrast-based motion illusions but experience more vection and less discomfort in a virtual environment compared to non-action video game players,” *J. Cogn. Enhanc.*, vol. 6, no. 1, pp. 3–19, 2022.

[65] T. T. Dündar, E. Cetinkaya, İ. Yurtsever, Ö. Uysal, and A. Aralaşmak, “Follow-up of high-grade glial tumor; Differentiation of posttreatment enhancement and tumoral enhancement by DCE-MR perfusion,” *Contrast Media Mol. Imaging*, vol., p. 6948422, 2022.

[66] A. Leftin and J. A. Koutcher, “Quantification of nanoparticle enhancement in polarized breast tumor macrophage deposits by spatial analysis of MRI and histological iron contrast using computer vision,” *Contrast Media Mol. Imaging*, vol. 2018, p. 3526438, 2018.

[67] B. I. Justusson, “Median filtering: Statistical properties,” in *Topics in Applied Physics*, Berlin/Heidelberg: Springer-Verlag, 2006, pp. 161–196.

[68] T. Loupas, W. N. McDicken, and P. L. Allan, “An adaptive weighted median filter for speckle suppression in medical ultrasonic images,” *IEEE Trans. Circuits Syst.*, vol. 36, no. 1, pp. 129–135, 1989.

[69] A. G. Zotin, “Fast algorithm of image enhancement based on multi-scale retinex,” *Int. j. reason.-based intell. syst.*, vol. 12, no. 2, p. 106, 2020.

[70] A. Zotin, “Fast algorithm of image enhancement based on multi-scale retinex,” *Procedia Comput. Sci.*, vol. 131, pp. 6–14, 2018.

[71] U. Nnolim, “Improved partial differential equation and fast approximation algorithm for hazy/underwater/dust storm image enhancement,” *Adv. Image Video Process.*, vol. 7, no. 3, 2019.

[72] C. Tang, U. F. von Lukas, M. Vahl, S. Wang, Y. Wang, and M. Tan, “Efficient underwater image and video enhancement based on Retinex,” *Signal Image Video Process.*, vol. 13, no. 5, pp. 1011–1018, 2019.

[73] C. Dietrich *et al.*, “How to perform Contrast-Enhanced Ultrasound (CEUS),” *Ultrasound Int. Open*, vol. 04, no. 01, pp. E2–E15, 2018.

[74] B. Leenknegt, Z. Harkanyi, A. Deganello, and P. S. Sidhu, “How to perform an intravenous contrast-enhanced ultrasound (CEUS) examination in a child. Methodology and technical considerations,” in *Contrast-Enhanced Ultrasound in Pediatric Imaging*, Cham: Springer International Publishing, 2021, pp. 39–45.

[75] J. Ma, X. Fan, S. X. Yang, X. Zhang, and X. Zhu, “Contrast limited adaptive histogram equalization-based fusion in YIQ and HSI color spaces for underwater image enhancement,” *Intern. J. Pattern Recognit. Artif. Intell.*, vol. 32, no. 07, p. 1854018, 2018.

[76] S. Ma, H. Ma, Y. Xu, S. Li, C. Lv, and M. Zhu, “A low-light sensor image enhancement algorithm based on HSI color model,” *Sensors (Basel)*, vol. 18, no. 10, p. 3583, 2018.
[77] X. Liu, Q. Xie, Q. Zhao, H. Wang, and D. Meng, “Low-light image enhancement by Retinex based algorithm unrolling and adjustment,” *arXiv [cs.CV]*, 2022.
[78] 国亮, “Research on low-light image enhancement algorithm based on retinex model,” *Comput. Sci. Appl.*, vol. 12, no. 04, pp. 895–903, 2022.
[79] J. Bennet, C. A. Ganaprakasam, and K. Arputharaj, “A discrete wavelet based feature extraction and hybrid classification technique for microarray data analysis,” *ScientificWorldJournal*, vol, p. 195470, 2014.
[80] M. Benco, R. Hudec, P. Kamencay, M. Zachariasova, and S. Matuska, “An advanced approach to extraction of colour texture features based on GLCM,” *Int. J. Adv. Robot. Syst.*, vol. 11, no. 7, p. 104, 2014.
[81] H. A. Dwaich and H. A. Abdulbaqi, “Signature texture features extraction using GLCM approach in android studio,” *J. Phys. Conf. Ser.*, vol. 1804, no. 1, p. 012043, 2021.
[82] S. A. Alazawi, N. M. Shati, and A. H. Abbas, “Texture features extraction based on GLCM for face retrieval system,” *Period. Eng. Nat. Sci. (PEN)*, vol. 7, no. 3, p. 1459, 2019.
[83] S. Shorman and S. Ali Pitchay, “Removing rain streaks from single images using total variation,” *Int. j. multimed. appl.*, vol. 10, no. 06, pp. 199–2017, 2018.
[84] T. Xi, J. Wang, Y. Han, C. Lin, and L. Ji, “Multiple butterfly recognition based on deep residual learning and image analysis,” *Entomol. Res.*, 2022.
[85] J. Wang, Z. Lin, Z. Feng, Y. Lu, and S. Shi, “Recognition of rock images and quantification of oil content using deep residual neural networks,” in *2021 4th International Conference on Signal Processing and Machine Learning*, 2021.
[86] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv [cs.CV]*, 2014.
[87] P. Duan, Z. Wang, B. Shi, O. Cossairt, T. Huang, and A. Katsaggelos, “Guided Event Filtering: Synergy between intensity images and neuromorphic events for high performance imaging,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PP, pp. 1–1, 2021.
[88] B. Ham, M. Cho, and J. Ponce, “Robust guided image filtering using nonconvex potentials,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 192–207, 2018.
[89] Y. Lin, D. Cao, and X. Zhou, “Adaptive infrared and visible image fusion method by using rolling guidance filter and saliency detection,” *Optik (Stuttg.)*, vol. 262, no. 169218, p. 169218, 2022.
[90] G. Prema and S. Arivazhagan, “Infrared and visible image fusion via multi-scale multi-layer rolling guidance filter,” *Pattern Anal. Appl.*, 2022.
[91] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “DeepFace: Closing the gap to human-level performance in face verification,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
[92] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv [cs.LG]*, 2015.

[93] M. Li, X. Cao, Q. Zhao, L. Zhang, and D. Meng, “Online rain/snow removal from surveillance videos,” *IEEE Trans. Image Process.*, vol. 30, pp. 2029–2044, 2021.

[94] T. Yan *et al.*, “Snow removal from light field images,” *IEEE Access*, vol. 7, pp. 164203–164215, 2019.

[95] H. Zhang and V. M. Patel, “Densely connected pyramid dehazing network,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[96] X. Hu, C.-W. Fu, L. Zhu, and P.-A. Heng, “Depth-attentional features for single-image rain removal,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[97] H. Zhang and V. M. Patel, “Density-aware single image DE-raining using a multi-stream dense network,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[98] Y. Liu, J. Ma, Y. Wang, and C. Zong, “A novel algorithm for detecting pedestrians on rainy image,” *Sensors (Basel)*, vol. 21, no. 1, p. 112, 2020.

[99] C. Peng *et al.*, “MegDet: A large mini-batch object detector,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[100] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[101] Deepika, “An overview of you only look once: Unified, real-time object detection,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 8, no. 6, pp. 607–609, 2020.

[102] H. Wang, Q. Xie, Q. Zhao, and D. Meng, “A model-driven deep neural network for single image rain removal,” *arXiv [eess.IV]*, 2020.

[103] A. Y. Virasova, D. I. Klimov, O. E. Khromov, I. R. Gubaidullin, and V. V. Oreshko, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Радиотехника*, pp. 115–126, 2021.

[104] L. Liu *et al.*, “Deep learning for generic object detection: A survey,” *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, 2020.

[105] K. Shi, Y. Zhu, and H. Pan, “A novel model based on deep learning for Pedestrian detection and Trajectory prediction,” in *IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, 2019.

[106] D. J. Jo, H. S. Yang, and Y. S. Moon, “Extended joint deep learning for pedestrian detection,” in *Proceedings of the 2nd World Congress on Electrical Engineering and Computer Systems and Science*, 2016.