This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2879473, IEEE Access

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier ...

Fuzzy connectives for efficient image reduction and speeding up image analysis

GLEB BELIAKOV[1] (Senior Member, IEEE), GITA DAS[1], HUY QUAN VU[2], TIM WILKIN[1] (Member, IEEE) and YONG XIANG[1] (Senior Member, IEEE)

[1]School of Information Technology, Deakin University, 221 Burwood Hwy, Burwood 3125, Australia (email: gleb@deakin.edu.au)
[2]Centre for Tourism and Regional Opportunities, School of Engineering and Technology, Central Queensland University, Australia (email: h.vu@cqu.edu.au)

Corresponding author: Gleb Beliakov (email: gleb@deakin.edu.au).

ABSTRACT We discuss non-monotone fuzzy connectives in large scale image processing. We present an image reduction algorithm capable of differentiating between fine image details and noise in the image, particularly salt and pepper noise. The reduction algorithm is based on mode-like averaging functions. We compare the performance of the proposed method to the alternative reduction methods on artificial images and on two case studies: content based image retrieval and pedestrian detection. Our algorithm improves the speed of the subsequently applied image analysis methods and allows efficient filtering of salt and pepper noise. Applications to on-board image recognition in autonomous robotic devices are envisaged.

INDEX TERMS Aggregation operators; Fuzzy connectives; Image reduction; Averaging; Mode.

I. INTRODUCTION

Modern data is captured much faster than it can be analysed. Novel applications of digital imaging demand super-fast image analysis and recognition. Self-driving vehicles, unmanned aerial vehicles (UAV) and autonomous robots are some examples where recognition (of small objects or pedestrians) from video footage must be performed in real time. UAVs and underwater robots are restricted in their computational capabilities, as more computer power means heavier battery and larger dimensions which are at a premium, and therefore sophisticated on-board image recognition algorithms are way too slow for autonomous operation of these devices.

One common issue in these applications is that the large amount of visual data captured by the high resolution cameras cannot be analysed on the low computing capacity devices by which the data is captured. The main reasons here are severe limitations on the cost, size, weight and power consumption. While image acquisition is performed at a linear rate (with image resolution), image analysis has a super-linear computational cost, and the computational bottleneck is thus inevitable. As the technology develops, more advanced processing hardware also means even higher resolution cameras, and the hence even higher demands on processing power.

Furthermore, the high rate of image acquisition leads to unavoidable noise in the image data, thermal noise being one of the problems here, but also missing data due to interference, radiation, etc. The likelihood of corrupted pixels in an image, while not very large per se, is not negligible and grows with image resolution. While dropping image resolution seems a feasible alternative here, that also degrades the quality of the subsequent analysis. One of the challenges in big data (at least in image processing) is that most of the data captured are uninteresting and are subsequently discarded, but have to be processed in order to identify the interesting parts, that need to be captured at the highest resolution available. Timely image analysis on board is also important for automation of the behaviour of autonomous devices, like identification of targets and obstacles.

This paper advocates the use of image reduction techniques based on granular computing, in particular on fuzzy sets aggregation techniques. The goal here is to artificially reduce image resolution to facilitate faster processing, for example on autonomous underwater or aerial vehicles, yet in such a way that the reduced image preserves the important characteristics of the original image and at the same time filters out Gaussian and non-Gaussian noise. Furthermore, we aim to achieve reduction while preserving very fine details of the high resolution image but distinguish them from noise. Often the objects of interest in the images are small, or have particular characteristics (like texture) that are not usually

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2879473, IEEE Access

preserved by the simple reduction methods. Finally, the linear time complexity of the reduction algorithm is paramount.

Thus image reduction consists in diminishing the resolution of the image while keeping as much information as possible from the original image [2], [31]. Image reduction (and the inverse process, image magnification) is a widely studied topic due to its applicability in mobile gadgets such as phones, PDAs or cameras, where the images must be visualized in small screens and the need of changing the size of the images without significantly changing their quality is very important.

In the literature there exist many different image reduction methods. Some of them consider the image to be reduced globally or in a transform domain [26], [29], [33]. Some others divide the image into pieces and act on each of them locally [2], [18], [28], [31], [34]. The latter procedure allows one to design algorithms which are very efficient in time (so the image reduction is performed at the same rate as image acquisition) and which preserve some of the specific properties of the images such as textures, edges, etc.

Our aim in this work is to design a reduction algorithm that, given an image, provides a new image of lower resolution that keeps the intensity properties of the original image, filters out non-additive noise and in addition preserves even very small details of the original image, such as 1-pixel wide lines. The latter is very challenging, as such details are easy to confuse with the noise.

We follow the approach presented in [2], [30], [31] where the concept of local image reduction operators was formalised. The input image is divided into possibly overlapping small rectangular blocks, and the intensities of the pixels in each block are combined to produce a representative value, which will be the intensity of one pixel in the output image. Thus we perform fusion of intensities of pixels in a block, which is then reduced to one pixel. Evidently, this process is computationally efficient and easily parallelisable.

The representative pixel value is calculated by using aggregation functions employed in the fuzzy sets theory, also called fuzzy connectives, and in particular averaging functions [1], [7], [23]. Typical averaging functions are the arithmetic mean and the median, and they are relied upon in the naive image reduction methods. We shall see later on that such averages result in oversmoothing of the images and losses of small details. We explore more sophisticated averages, such as penalty-based and mode-like functions presented in [1], [3], [5], [42] and we show their superior performance in image reduction tasks, especially in the presence of noise.

In order to study the use of sophisticated reduction operators, we carry out several experimental studies. Firstly, we perform laboratory experiments using artificially generated images, in order to demonstrate the capability of the selected averaging functions to differentiate between the noise and small image details.

Then, we perform experiments with real-world images on two case studies. The first case study is image retrieval [17], [36], [37]. We use state-of-the-art retrieval methods in order to query image databases and select images matching a particular query image. In this task the vector of image features is computed and matched against feature vectors in the image database. We analyse the performance of image retrieval algorithms (in terms of the accuracy and CPU time) for full scale and reduced images as a function of image reduction method employed.

The second case study is pedestrian identification [21]. This problem is particularly important for the development of self-driving vehicles. It is conjectured that the expensive equipment currently used on prototypes of such vehicles can be replaced with inexpensive small cameras coupled with pedestrian identification algorithms. Here again, we analyse the performance of pedestrian identification algorithms for full-scale and reduced images in the presence of noise in the data, as a function of image reduction method.

In both cases we use off-the-shelf third party image analysis algorithms and apply them to the full-scale and reduced images. Evidently, the analysis algorithm will work much faster on the reduced images. The question is: what are the losses in quality of the output (e.g., accuracy), and whether such losses are justified by the gains in the speed of processing.

In both case studies we contaminate the query images with different types of noise: Gaussian, speckle, and salt and pepper noises, and analyse the accuracy of the recognition methods as a function of noise and the reduction method applied.

The paper is structured as follows. Section II gives preliminary definitions. Section III presents mode-like averaging functions based on minimisation of various penalties. Section IV outlines the image reduction algorithm. Section V details our the experimental analysis. Section VI concludes.

II. PRELIMINARY DEFINITIONS

A. IMAGE REDUCTION

We now formulate the problem of image reduction based on local, block-based reduction operators. Local operators based on aggregation functions have been shown to be both effective and fast, and they easily admit parallel implementations [2], [31], [32]. In this approach an image of size $M \times N$ is subdivided into non-overlapping blocks of size $m \times n$, and each block is aggregated to generate a single value representative of the original data, as illustrated on Figure 1. This value becomes the intensity of a single pixel within the reduced image. The original image is reduced to the size $M' \times N' = \lfloor \frac{M}{m} \rfloor \times \lfloor \frac{N}{n} \rfloor$. Image reduction should not be confused with image compression. The former attempts to represent the image using a smaller number of pixels having the same data precision, whereas the latter attempts to represent the original image data using a lower precision. Image reduction is frequently employed when displaying images on smaller screens, when performing operations such as zooming (dilation), or to speed up subsequent image analysis tasks, such as feature detection or recognition.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2879473, IEEE Access

The core of block-based reduction algorithms is a function which generates a representative intensity value for a given block of pixels. Such functions require the properties of averaging functions, such that the output is within the range of the intensities of the input pixels, and is also idempotent. This latter property ensures that if the block is of uniform intensity, its representative value is exactly the same as the intensity of all input pixels.

There are several alternative methods of image reduction. The interpolation methods, like bilinear and bicubic interpolation, build a piecewise polynomial model of the image, and then compute the pixel values of the reduced image by using such an interpolant. Fuzzy transform methods (F-transform) [18], [32]–[34] have also been applied to image reduction and will be used here for benchmarking.

Color image reduction is typically performed componentwise, by calculating intensities of each color channel separately. However that might lead to pixel colors not present in the original image. We show later how this can be avoided by using penalty-based approach.

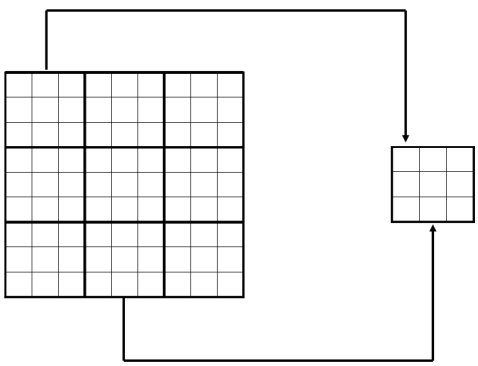


FIGURE 1: A scheme for image reduction based on representative values within 3×3 blocks.

B. AVERAGING FUNCTIONS

The class of functions with the above mentioned requirements is known as means (or averages), and is well studied, in particular from the point of view of fuzzy connectives [1], [7], [11]. The simplest examples are the weighted arithmetic mean and the median, which play an important role in Gaussian and median filtering. All suitable averaging functions can be obtained by using penalty-based approach [13], wherein a certain penalty for intensity deviations is minimized. A comparative analysis of various penalty-based averages in this context was presented in [2], [12], [31].

Typically, averaging functions satisfy another condition, that of monotonicity [7], [11]. If any of the input values increases, the output cannot decrease. This condition is useful when the data are noiseless, however, when the data may be contaminated by noise, monotonicity is not desirable. An increase or decrease of the intensity of a pixel to an extreme (or away from the most prevalent value in a region) is plausibly indicating that it is corrupted and thus should be disregarded in the average. Permitting outlying pixel intensity to affect the average of the image block leads to a degradation of performance in tasks such as filtering and image reduction.

Non-monotone averages have been treated in [4], [5], [8], [9], [41], [42], and they also play a significant role as robust estimators of location [35]. Typical examples here are the shorth and the least trimmed squares (LTS) estimator. Non-monotone averages can be represented as penalty-based functions [1], [13], and it is within this context that we wish to design penalties that account for the spatial organisation of pixels. The subsequent averaging functions may then be applied within block-based image reduction of noisy images.

Formally, following [2], [13], if the input intensities in a block are represented by $\mathbf{x} \in \mathbb{I}^k$, $k = m \times n$, the average intensity is found by solving the minimisation problem

$$y = f(\mathbf{x}) = \arg\min_y \mathcal{P}(\mathbf{x}, y),$$

where $\mathcal{P} : \mathbb{I}^{k+1} \to \mathbb{R}$ is a penalty function satisfying the conditions:

1) $\mathcal{P}(\mathbf{x}, y) \geq c \quad \forall \mathbf{x} \in \mathbb{I}^k,\ y \in \mathbb{I}$;
2) $\mathcal{P}(\mathbf{x}, y) = c$ if and only if all $x_i = y$

for some constant $c \in \mathbb{R}$ and any closed, non-empty interval $\mathbb{I}$. That is, the average is the value y which minimises the total penalty $\mathcal{P}$ for disagreement between that value and the individual pixel intensities x_i. It is known that all idempotent functions can be represented in this way [1], [13].

For example, the arithmetic mean is obtained by using squares of the differences,

$$\mathcal{P}(\mathbf{x}, y) = \sum_{i=1}^{k} (x_i - y)^2,$$

whereas the median is obtained by using the absolute differences

$$\mathcal{P}(\mathbf{x}, y) = \sum_{i=1}^{k} |x_i - y|.$$

III. MODE-LIKE AVERAGING

It is known that the arithmetic mean is not robust against outliers [35]. Just one extreme value of an input (like when we have the salt and pepper noise) drives the output in its direction away from the the main group of uncontaminated inputs. The median is robust, as its value is not affected by the outliers, but it discards all but one (central) input. There are other robust estimators of location [35] such as the shorth and the least trimmed squares estimator, which all belong to the class of weakly monotone averages [1], [42]. These averages model the majority: the output is always a value representing the majority of inputs, which are considered uncontaminated by noise.

However, in this study we want to model the "vocal minority" - a subset of inputs with consistently close values, which may or may not represent the majority. For example, a fragment of a thin one pixel-wide line on a not perfectly uniform background consists of just a few pixels of close

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2879473, IEEE Access

intensities, whereas the majority of pixels in a block belong to the background, but their intensities are more spread. To preserve such fine lines in the reduction process we need to select a compact cluster of inputs and discard the majority which are not closely aligned with each other.

The mode function selects the most frequent input, and if we have a perfect cluster of inputs that share the same value, the mode would identify such a value. Mode-seeking algorithms are known in image filtering [14], [40]. In reality, however, a cluster of pixels intensities will not have one common value, and in fact all inputs might be distinct, in which case there is no unique mode. For this reason below we define a mode-like average which mimics the mode when the inputs are distinct. For this we use penalty based approach [13].

Unlike in [2], [12], [13], where $\mathcal{P}$ was required to be quasiconvex to guarantee a unique minimum, and also to ensure continuity of f, we do not impose this condition as we are interested in non-monotone averages. In [41] a penalty function based on a weighted sum of intensity-based partial penalties was given by

$$\mathcal{P}(\mathbf{x}, y) = \sum_{i=1}^{k} w_i(y)\rho(x_i, y) \tag{1}$$

where

$$\rho(x_i, y) = \begin{cases} r_{(j)} & r_{(j)} < \tau, \\ \beta_\tau & r_{(j)} \geq \tau. \end{cases} \tag{2}$$

$$\tau = \alpha \max(\epsilon, r_{(t)}) \text{ and } \alpha > 0, 0 \leq \beta \leq 1, 2 \leq t \leq k.$$

The parameters of this model are described below.

Given $r_i = |x_i - y|$ then $r_{(j)}$ denotes the jth smallest element of the set of ordered (ascending) values of r_i, the absolute deviations from the aggregate value y. This function generates a mode-like non-monotonic average of the input vector $\mathbf{x}$, and it was shown to outperform other monotonic and non-monotonic block-based reduction operators when applied to images corrupted by speckle or impulse noise.

The function ρ favours compact clusters of intensity values in $\mathbf{x}$ by assigning smaller penalties to inputs closer to the proposed output. Thus the value y that minimises $\mathcal{P}$ is a value representative of the most compact cluster within $\mathbf{x}$. The parameter α tunes the cluster boundary as a function of the minimum non-zero intensity difference between inputs, and β is a fixed constant penalty for non-cluster members. Thus $\mathcal{P}$ attempts to minimise the number of outliers while also trying to identify the most compact cluster. This averaging method is called $Pmode$.

However, ρ does not take into account spacial information of the pixels; i.e., it does not distinguish between a geometrically compact group of pixels having similar intensity and similar pixels scattered within a given block. To account for this the weights w_i should depend on the position of the pixels within the block. In [41] this was achieved using a normalised distance,

$$w_i(y) = \frac{d(x_i, y)}{\sum_{i=1}^{k} d(x_i, y)}, \forall y = x_j \in \{x_1, ..., x_k\}. \tag{3}$$

This function arose from the additional constraint that the average must also be a locally internal function (i.e., the output should be one of the inputs), which is a reasonable requirement in image reduction tasks. A more sophisticated weighting function was presented in [6], which is based on fuzzy measures. That function favours larger clusters of pixel intensities, which are also compact in the geometrical sense. For example three neighboring pixels with similar intensities are favored with respect to pixels spread over the image block. The penalty in [6] which accounts for spatial organisation of pixels is expressed as

$$\mathcal{P}(\mathbf{x}, y) = w(\mathcal{A}) \sum_{i=1}^{k} \rho(x_i, y), \tag{4}$$

where cluster-based weights $w(\mathcal{A})$ replace the individual pixel weights $w_i(y)$ in (1). Here $w(\mathcal{A}) = 2 - v(\mathcal{A}) \geq 1$, v is a fuzzy measure computed by one of three approaches in [6] and $\mathcal{A}$ denotes the subset of pixels having intensities that satisfy $r_{(k)} < \tau$ as per Eqn. (2). This average is called the $Cmode$. The values of w are precomputed and thus do not affect the run time of the $Cmode$ algorithm.

IV. REDUCTION ALGORITHM

The overall block-based image reduction algorithm is as follows.

ALGORITHM 1: Image reduction algorithm.

Inputs: A (original image as an $M \times N$ array), M, N (dimensions), m, n (reduction ratios)
Output: B (the reduced image of size $M' \times N'$)

1. $M' = \lfloor \frac{M}{m} \rfloor$ $N' = \lfloor \frac{N}{n} \rfloor$
2. Select an averaging function F
3. For each (non-overlapping) block of A of size $m \times n$ denoted by C
   3.1 $c = F(C)$;
4. Build the reduced image B from the outputs of Step 3.1.

In our study we used several different averaging functions as F, the $Pmode$ and $Cmode$ functions as described in the previous section, shorth, and the LTS estimator [35] as examples of robust estimators, and the arithmetic mean and the median for comparison purposes. We also used alternative reduction methods based on bilinear and bicubic interpolation and Lanczos method (available in Matlab Image Processing toolbox), as well as based on the F-transform [32], [33].

The minimization of the penalty function in the $Pmode$ and $Cmode$ functions was performed over the set of input intensities, which ensured the output did not contain pixel intensities not present in the input image. The weighting

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2879473, IEEE Access

TABLE 1: List of features reflecting image characteristics.

Feature	Description	Reference
F_1	Entropy	[22]
F_2	Standard deviation	Matlab
F_3	Skewness	Matlab
F_4	Kurtosis	Matlab
F_5	Texture based Wavelet transform	[16]
F_6	Noise Level	[27]
F_7	Jpeg quality score	[39]
F_8	Sharpness index	[10]
F_9	Contrast	[24], [38]
F_{10}	Cluster Shade	[38]
F_{11}	Dissimilarity	[38]
F_{12}	Sum of squares: Variance	[24]

functions w were precomputed. Note that in the case of color images, it is not necessary to perform reduction for each color channel independently. Penalty minimisation can be performed on a product lattice of color intensities as presented in [2].

V. EXPERIMENTAL ANALYSIS

A. FINE DETAIL PRESERVATION

In the first series of experiments we used synthetic grayscale images called respectively lines and circles (Figures 2-3). The lines and circles are one pixel wide, nearly white on non-uniform gray background. The intensities of pixels of the lines and circles are not constant but vary within 10% of the range of intensities (the standard 0–255 range), whereas the intensities of the gray background vary within 30% of the range of intensities. Subsequently salt and pepper noise was added to the image, changing 0.5% of pixels to black or white.

Our aim here is to establish the robustness of the reduction methods to the salt and pepper noise and their ability to preserve visible fine line structures. In addition to visual inspection, which can be done using Figures 2-3), we also calculated various quantitative characteristics of the original image, noisy image and the reduced images. The image characteristics are presented in Table 1, together with the references to their origins.

It is assumed that a good reduction method would preserve the numerical characteristics of an image as much as possible, and would not allow noise in the image alter these characteristics. Therefore we computed the mentioned characteristics for the noiseless image, for the noisy image for comparison, and then for reduced noisy images. The images were reduced by one of the mentioned methods by a factor of 9 (both $m, n = 3$ in Algorithm 1).

Figures 2-3 reveal that the mean block-reduction and bilinear interpolation methods preserve the lines structure, but smoothen the lines. Indeed, the pixel intensity of the visible lines is only a fraction of that in the original image, which is averaged with the intensity of the background. The median and the shorth functions, while robust to the salt and pepper noise, also remove the lines and circles from the image, which is unsatisfactory. The $Pmode$ and $Cmode$ functions produce the lines and the circles with their original intensity and filter most but some noise.

In Tables 2-3 the results for the original and noisy images are in the first two rows of these tables. The rest of the rows refer to reduction of the noisy images, but are compared to the original image, as we aim to preserve its characteristics despite the presence of noise. The values which match the best those for the original image are in boldface, whereas the differences, in the second half of each table, are illustrative in comparison to the values in row 1, as they indicate by how much the best performing method is better than the alternatives. Drastic differences indicate that one method was significantly better than the others. We see that in the Lines image, all the characteristics are better preserved when using the $Pmode$ and $Cmode$ algorithms. In case of the Circles image, some characteristics are better preserved using $Pmode$ and $Cmode$, but there are some which are preserved better with more smoothing.

B. IMAGE RETRIEVAL

Content-Based Image Retrieval (CBIR), where an image is represented by its visual and semantic contents, is crucial to successfully retrieve images from huge databases [36], [37]. A feature vector is extracted for each query image and then compared to the feature vectors in the database. Based on the matching of feature vectors, the most similar images to the query are retrieved.

While extracting feature vectors from the database can be done offline, and hence computation time is not that critical, extraction of the features from the query images is usually performed online and it is time-critical. In addition query images could be noisy, and therefore noise filtering is required. Performing noise-cancelling reduction of the query images allows to significantly speed up the whole CBIR process.

We studied how image reduction affects the precision of the image retrieval process measured as

$$precision = \frac{\text{Number of relevant retrieved images}}{\text{Number of retrieved images}}.$$

We have used the scope value (i.e., the number of retrieved images presented to the user) of 20.

We have chosen Wang database (http://wang.ist.psu.edu/docs/related), a benchmark database that has been widely used in CBIR research community. The Wang database is a subset of 1,000 images of the Corel stock photo database which has been manually selected and which form 10 classes (contains Africa, Beach, Buses, Dinosaurs, Elephants, Flowers, Food, Horses, Monuments, Mountains) of 100 images each. Each image is of size 256 x 384. All images have been converted to `png` from `jpeg` format (using Format Factory software) to form the original dataset.

To model a scenario when the CBIR system is presented with noisy images, noise (as indicated in Table 4) was added to query images, and then the images were either reduced by

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2879473, IEEE Access

one of the mentioned methods by a factor of 9 (both $m, n = 3$ in Algorithm 1), or left unchanged (for comparison).

A compact 25-dimensional feature vector using the elements of Colour Co-occurrence Matrices (CCM) in Hue, Saturation, Value (H,S,V=16,3,3) space [15] was then extracted from the query images and compared against feature vectors in the original, noiseless database. The L1-norm was used to measure similarity between query and database images.

As expected, we obtained a reduction in computation time by a factor of 7.8 when using reduced images (total time 242.49 sec for 1000 full images vs 30.74 sec for reduced images). The retrieval time was not affected by reduction.

The results are presented in Table 4. We can observe that compared to clean original images (the gold standard in this case) all reduction methods have shown a degradation in retrieval accuracy. This reduction is least for our $Pmode$ and $Cmode$ algorithms. On the other hand, when noisy query images were presented, in all cases reduction resulted in improvements to the retrieval precision, attributed to smoothing filtering effect of reduction.

The algorithms performed differently depending on the type of noise in the images. For Gaussian and speckle noise with variances indicated in Table 4, block-based reduction by using the mean function delivered the best results. This is expected, as the arithmetic mean is precisely the minimizer of the sum of squared deviations, which is the maximum likelihood estimator under Gaussian noise assumption [13]. F-transform also delivered similar precision. For speckle noise with small variance, bicubic reduction and $Cmode$ resulted in good precision, but so was the benchmark - using noisy original size images. This means that filtering in the reduction process did not play a major role. For speckle noise with larger variance the arithmetic mean-based reduction delivered better results.

For the salt and pepper noise the proposed $Pmode$ and $Cmode$ algorithms were clear winners. These two methods also performed well for clean images. The median filtering, which is usually used for this type of noise, was not as effective, most likely because of reduction in image contrast and other related features. The mean-based filtering was unable to remove the outliers. In contrast, $Pmode$ and $Cmode$ did not affect the features used for retrieval, yet were able to successfully filter out the salt and pepper noise.

C. PEDESTRIAN DETECTION

Pedestrian detection is an essential task in Computer Vision [21], where the quality and speed of the detector is of importance. The use of reduced test images can contribute towards less computation time and provide better detection accuracy in noisy environment.

In this study we looked at the performance of the Aggregate Channel Features (ACF) object detector [20], [21], which is a fast and effective sliding window detector, on the original, noisy and reduced noisy images. Computations were performed using Piotr's Computer Vision Matlab Toolbox [19]. We used the INRIA pedestrian dataset [25], one of the most popular static pedestrian detection datasets. It contains high resolution (varying from 1280x960 to 531x333) video (`.seq`) files which are converted into image frames. There are 614 positive training images along with ground truth Bounding Box(BB) and 1218 negative images with no pedestrians. For testing we used 288 positive images along with ground truth BBs. Ground truth BBs are scaled down by 3 when reduced test images are used. The test images were added different types of noise: Salt and pepper, Gaussian, speckle, and a mixture of Gaussian, speckle and salt and pepper noises added one after another in that order.

For reduced test images the detector was modified to work with downscaled images. The training images were not reduced. Features for training images are computed once only, and the features at other scales can be approximated. This makes the detector fast.

The following reduction operators were used: Mean, Median, Pmode, Cmode, Bicubic and F-transform. Other reduction methods did not show any significant difference in performance to the existing methods used for benchmarking, and hence the results are omitted.

As expected, the use of reduced test images resulted in significant reduction in the detection time, by a factor of 9.5 (7.95 frames per second vs. 0.83 frames per second). The accuracy of the detector was measured through the Log-average Miss Rate [20], [21], where the lower values mean better accuracy.

The results are presented in Table 5. As expected, added noise degrades the performance of the detector, often significantly (data row 1). The reduction of clean test images did not result in a drastic decrease in performance for the existing reduction methods, but led to some decrease in performance for the proposed $Pmode$ and $Cmode$ (data column 2).

However, reduction of noisy images resulted in a better performance than with non-reduced images. For Gaussian and speckle noise, as well as mixtures with these noises, the arithmetic mean-based reduction was the best approach, followed by F-transform. This is expected as the arithmetic mean effectively filters out Gaussian noise. For the salt and pepper noise the $Pmode$ and $Cmode$ have shown superior performance, very significant at the 5% noise level. Surprisingly, the median-based reduction led to very poor performance of the detector.

Figure 4 shows one example of a frame with pedestrians and the bounding boxes found by the ACF algorithm. The ROC curves for different reduction methods are presented in Figure 5.

VI. CONCLUSION

This paper discussed the use of fuzzy connectives in processing large image streams as a tool to overcome computational bottlenecks. Local image reduction is computationally efficient, easily parallelisable method of reducing and filtering images at approximately the same rate of image acquisition. The subsequent image analysis can then be performed at a much faster pace. To preserve the accuracy of image analysis

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2879473, IEEE Access

algorithms it is important to preserve important features of an image and differentiate between noise and fine details.

In this article we applied mode-like averaging functions as local reduction operators. We found that such functions are better at preserving the essential characteristics of an image and can differentiate between salt and pepper noise and 1-pixel wide lines and curves. The traditional reduction techniques often oversmooth the images, but are better dealing with Gaussian and speckle noises. We also conducted two case studies, content based image retrieval and pedestrian detection, where we applied third party image analysis algorithms to the original, noisy and reduced images.

We found that the proposed $Pmode$ and $Cmode$ algorithms are better at preserving accuracy of the subsequently applied methods in the presence of salt and pepper noise. In all cases significant acceleration of the queries (in CBIR) and detection (Pedestrian) was achieved by using the reduced images, and the accuracy of these methods was better when compared to the large-sized but noisy images. The question as to which reduction method to use depends much on the type of noise in the images. For salt and pepper noise we recommend the mode-like reduction, whereas for Gaussian and speckle noise mean-based local reduction and F-transform methods are preferable.

REFERENCES

[1] G. Beliakov, H. Bustince, and T. Calvo. A Practical Guide to Averaging Functions. Studies in Fuzziness and Soft Computing. Springer-Verlag, Berlin, Heidelberg, Berlin, 2016.

[2] G. Beliakov, H. Bustince, and D. Paternain. Image reduction using means on discrete product lattices. IEEE Trans. on Image Processing, 21:1070–1083, 2012.

[3] G. Beliakov, T. Calvo, and T. Wilkin. Three types of monotonicity of averaging aggregation. Knowledge-Based Systems, 72:114–122, 2014.

[4] G. Beliakov, T. Calvo, and T. Wilkin. On the weak monotonicity of Gini means and other mixture functions. Information Sciences, 300:70–84, 2015.

[5] G. Beliakov, S. James, T Wilkin, and T. Calvo. Robustifying OWA operators for aggregating data with outliers. IEEE Trans. on Fuzzy Systems, 26:1823–1832, 2017.

[6] G. Beliakov, G. Li, H.Q. Vu, and T. Wilkin. Characterizing compactness of geometrical clusters using fuzzy measures. IEEE Trans. on Fuzzy Systems, 23:1030–1043, 2015.

[7] G. Beliakov, A. Pradera, and T. Calvo. Aggregation Functions: A Guide for Practitioners, volume 221 of Studies in Fuzziness and Soft Computing. Springer-Verlag, Berlin, 2007.

[8] G. Beliakov and J. Spirkova. Weak monotonicity of Lehmer and Gini means. Fuzzy Sets and Systems, 299:26–40, 2016.

[9] G. Beliakov and T. Wilkin. On some properties of weighted averaging with variable weights. Information Sciences, 281:1–7, 2014.

[10] G. Blanchet and L. Moisan. An explicit sharpness index related to global phase coherence. In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, page 1065 Ű 1068, Kyoto, Japan, March 2012.

[11] P.S. Bullen. Handbook of Means and Their Inequalities. Kluwer, Dordrecht, 2003.

[12] H. Bustince, G. Beliakov, G. Pereira Dimuro, B. Bedregal, and R. Mesiar. On the definition of penalty functions in data aggregation. Fuzzy Sets and Systems, 323:1–18, 2017.

[13] T. Calvo and G. Beliakov. Aggregation functions based on penalties. Fuzzy Sets and Systems, 161:1420–1436, 2010.

[14] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Recognition and Machine Intelligence, 24:603–619, 2002.

[15] G. Das and S. Ray. Feature Re-weighting in Content-Based Image Retrieval. In Proceedings of 5th Intl Conf on Image and Video Retrieval, Lecture Notes in Computer Science 4071 Springer 2006, pages 194–200, Tempe, AZ, USA, July 13-15 2006.

[16] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Studying aesthetics in photographic images using a computational approach. In Proc. of European Conference on Computer Vision, pages 288–301, Graz, Austria, May 2006.

[17] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys, 40:Article 5, 2008.

[18] F. Di Martino, V. Loia, I. Perfilieva, and S. Sessa. An image coding/decoding method based on direct and inverse fuzzy transforms. International Journal of Approximate Reasoning, 48:110–131, 2008.

[19] P. Dollár. Piotr's Computer Vision Matlab Toolbox (PMT). http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html.

[20] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. IEEE Transactions on Pattern Recognition and Machine Intelligence, 36:1532 – 1545, 2014.

[21] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. IEEE Transactions on Pattern Recognition and Machine Intelligence, 34:743 – 761, 2012.

[22] R.C. Gonzalez, R.E. Woods, and S.L. Eddins. Digital Image Processing Using MATLAB, chapter 11. Prentice Hall, New Jersey, 2003.

[23] M. Grabisch, J.-L. Marichal, R. Mesiar, and E. Pap. Aggregation Functions. Encyclopedia of Mathematics and Its Foundations. Cambridge University Press, 2009.

[24] R. M. Haralick, K. S. Shanmugam, and I. Dinstein. Textural features for image classification. IEEE Trans. on Systems, Man and Cybernetics, 3:610–620, 1973.

[25] INRIA. Inria pedestrian dataset. Available from: http://pascal.inrialpes.fr/data/human/, accessed through http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/datasets/INRIA/, 2005.

[26] H. Kirshner and M. Porat. On the role of exponential splines in image interpolation. IEEE Transactions on Image Processing, 18:2198–2208, 2009.

[27] X. Liu, M. Tanaka, and M. Okutomi. Noise level estimation using weak textured patches of a single noisy image. In Proc. of the 19th IEEE International Conference on Image Processing, pages 665–668, Orlando, FL, 2012.

[28] V. Loia and S. Sessa. Fuzzy relation equations for coding/decoding processes of images and videos. Information Sciences, 171:145–172, 2005.

[29] H. Nobuhara, K. Hirota, S. Sessa, and W. Pedrycz. Efficient decomposition methods of fuzzy relation and their application to image decomposition. Applied Soft Computing, 5:399–408, 2005.

[30] D. Paternain, H. Bustince, J. Fernández, G. Beliakov, and R. Mesiar. Some averaging functions in image reduction. In Trends in applied intelligent systems, pages 399–408. Springer Berlin Heidelberg, 2010.

[31] D. Paternain, J. Fernandez, H. Bustince, R. Mesiar, and G. Beliakov. Construction of image reduction operators using averaging aggregation functions. Fuzzy sets and sstems, pages 1–25, 2014.

[32] D. Paternain, A. Jurio, R. Mesiar, G. Beliakov, and H. Bustince. Image reduction operators as aggregation functions: Fuzzy transform and undersampling. Advances in Soft Computing, 228:405–414, Nov 2013.

[33] I. Perfilieva. Fuzzy transforms: Theory and applications. Fuzzy Sets and Systems, 157:993–1023, 2006.

[34] I. Perfilieva and P. Vlašánek. Image reconstruction by means of f-transform. Know.-Based Syst., 70(C):55–63, November 2014.

[35] P.J. Rousseeuw and A.M. Leroy. Robust Regression and Outlier Detection. Wiley, New York, 2003.

[36] Y. Rui, T. S. Huang, and Sh-F. Chang. Image retrieval: Current techniques, promising directions, and open issues. Journal of visual communication and image representation, 10:39–62, 1999.

[37] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence,, 22:1349–1380, 2000.

[38] L. Soh and C. Tsatsoulis. Texture analysis of sar sea ice imagery using gray level co-occurrence matrices. IEEE Transactions on Geoscience and Remote Sensing, 37(2), 1999.

[39] Z. Wang, H. R. Sheikh, and A. C. Bovik. No-reference perceptual quality assessment of jpeg compressed images. In Proceedings of International Conference on Image Processing, pages 477–480, Rochester, New York, USA, September 2002.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2879473, IEEE Access

[40] J.van de Weijer and R.van den Boomgaard. Local mode filtering. In Computer Vision and Pattern Recognition, volume 2, pages 428–433. IEEE Computer Society, 2001.

[41] T. Wilkin. Image reduction operators based on non-monotonic averaging functions. In Proceedings of the 10th IEEE Int. Conf. on Fuzzy Systems, DOI 10.1109/FUZZ-IEEE.2013.6622458, Hyderabad, India, 2013.

[42] T. Wilkin and G. Beliakov. Weakly monotone averaging functions. Int. J. Intelligent Systems, 30:144–169, 2015.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2879473, IEEE Access

TABLE 2: Features Extracted from the Lines Images.

Images	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
Original	0.055	37.328	10.467	2.506	-0.053	19.352	49.828	1.610	1.462	29.960	0.769	6.827
Noise	0.073	38.595	10.705	2.548	-0.063	19.535	50.997	1.786	1.630	30.359	0.796	6.951
LTS	0.000	11.779	2.722	0.012	-0.004	11.179	22.021	0.209	0.373	0.056	0.340	4.391
MEAN	0.000	18.213	4.531	1.568	0.108	6.927	28.412	0.389	0.280	2.944	0.226	5.560
MEDIAN	0.000	9.654	3.878	0.516	0.000	8.099	14.009	0.391	0.191	0.195	0.187	4.450
BICUBIC	0.000	16.353	3.988	1.369	0.049	3.520	27.032	0.423	0.174	1.815	0.174	5.411
BILINEAR	0.000	14.601	4.247	1.438	0.057	2.817	22.553	0.432	0.149	1.307	0.149	5.260
LANCZOS2	0.000	16.368	3.931	1.352	0.039	3.509	27.031	0.423	0.172	1.795	0.172	5.406
LANCZOS3	0.000	17.350	3.692	1.248	0.045	3.918	29.593	0.427	0.188	1.980	0.187	5.501
SHORTH	0.000	11.670	2.796	0.022	-0.034	11.124	21.605	0.227	0.360	0.043	0.330	4.349
PMODE	**0.027**	38.230	**11.475**	**2.965**	**0.395**	**15.862**	50.844	**2.600**	2.429	21.491	0.754	7.153
CMODE	0.019	**47.286**	6.884	2.277	-0.011	11.507	**58.172**	1.106	**2.758**	**41.449**	**0.794**	**8.960**
Difference from the Original Image												
LTS	0.055	25.549	7.745	2.494	0.049	8.173	27.807	1.401	1.089	29.904	0.429	2.436
MEAN	0.055	19.115	5.936	0.938	0.161	12.425	21.416	1.221	1.182	27.016	0.543	1.267
MEDIAN	0.055	27.674	6.589	1.99	0.053	11.253	35.819	1.219	1.271	29.765	0.582	2.377
BICUBIC	0.055	20.975	6.479	1.137	0.102	15.832	22.796	1.187	1.288	28.145	0.595	1.416
BILINEAR	0.055	22.727	6.22	1.068	0.110	16.535	27.275	1.178	1.313	28.653	0.620	1.567
LANCZOS2	0.055	20.960	6.536	1.154	0.092	15.843	22.797	1.187	1.290	28.165	0.597	1.421
LANCZOS3	0.055	19.978	6.775	1.258	0.098	15.434	20.235	1.183	1.274	27.98	0.582	1.326
SHORTH	0.055	25.658	7.671	2.484	*0.019*	8.228	28.223	1.383	1.102	29.917	0.439	2.478
PMODE	*0.028*	*0.902*	*1.008*	0.459	0.448	*3.490*	*1.016*	0.990	*0.967*	*8.469*	*0.015*	*0.326*
CMODE	0.036	9.958	3.583	*0.229*	0.042	7.845	8.344	*0.504*	1.296	11.489	0.025	2.133

TABLE 3: Features Extracted from the Circles Images.

Images	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
Original	0.056	41.030	8.822	2.350	-0.065	10.500	65.013	511.976	2.108	28.606	0.776	7.460
Noise	0.073	42.201	8.951	2.371	-0.073	11.072	66.136	484.285	2.273	29.042	0.804	7.588
LTS	0.000	24.912	24.181	3.918	-0.155	14.869	35.546	47.353	1.223	6.960	0.567	5.344
MEAN	0.000	22.652	5.963	1.710	-0.036	7.823	37.543	10.286	0.765	2.801	0.474	6.060
MEDIAN	0.000	23.397	**27.693**	**4.417**	-0.097	10.801	31.943	**67.692**	1.033	7.124	0.448	5.448
BICUBIC	0.000	21.047	5.141	1.536	-0.028	3.945	35.892	8.298	0.565	2.521	0.394	5.978
BILINEAR	0.000	18.513	5.310	1.569	-0.024	3.163	31.773	9.063	0.436	2.076	0.328	5.787
LANCZOS2	0.000	21.075	5.098	1.526	-0.027	4.142	35.872	8.173	0.567	2.509	0.395	5.981
LANCZOS3	0.000	22.659	4.842	1.438	-0.029	4.743	38.523	7.915	0.674	2.641	0.454	6.093
SHORTH	0.000	24.899	24.365	3.943	-0.177	14.716	35.313	49.310	1.225	7.011	0.565	5.291
PMODE	0.039	42.978	8.492	2.454	-0.186	**28.973**	60.136	25.911	3.248	22.805	0.961	7.941
CMODE	**0.046**	**46.551**	6.952	2.189	**0.105**	24.677	**64.949**	24.982	**3.550**	**28.367**	**1.013**	**8.680**
Difference from the Original Image												
LTS	0.056	16.118	15.359	1.568	0.090	4.369	29.467	464.623	0.885	21.646	0.209	2.116
MEAN	0.056	18.378	2.859	0.640	*0.029*	2.677	27.470	501.690	1.343	25.805	0.302	1.400
MEDIAN	0.056	17.633	18.871	2.067	0.032	*0.301*	33.070	*444.284*	1.075	21.482	0.328	2.012
BICUBIC	0.056	19.983	3.681	0.814	0.037	6.555	29.121	503.678	1.543	26.085	0.382	1.482
BILINEAR	0.056	22.517	3.512	0.781	0.041	7.337	33.240	502.913	1.672	26.530	0.448	1.673
LANCZOS2	0.056	19.955	3.724	0.824	0.038	6.358	29.141	503.803	1.541	26.097	0.381	1.479
LANCZOS3	0.056	18.371	3.980	0.912	0.036	5.757	26.490	504.061	1.434	25.965	0.322	1.367
SHORTH	0.056	16.131	15.543	1.593	0.112	4.216	29.700	462.666	*0.883*	21.595	0.211	2.169
PMODE	0.017	*1.948*	*0.330*	0.104	0.121	18.473	4.877	486.065	1.140	5.801	*0.185*	*0.481*
CMODE	*0.010*	5.521	1.870	*0.161*	0.170	14.177	*0.064*	486.994	1.442	*0.239*	0.237	1.220

TABLE 4: Precison (%) at scope 20 with query images, both clean and noisy, original and reduced by the indicated method. The noise level is indicated by the percentage (S&P) and variance (Gaussian and Speckle). The best precision is indicated in boldface.

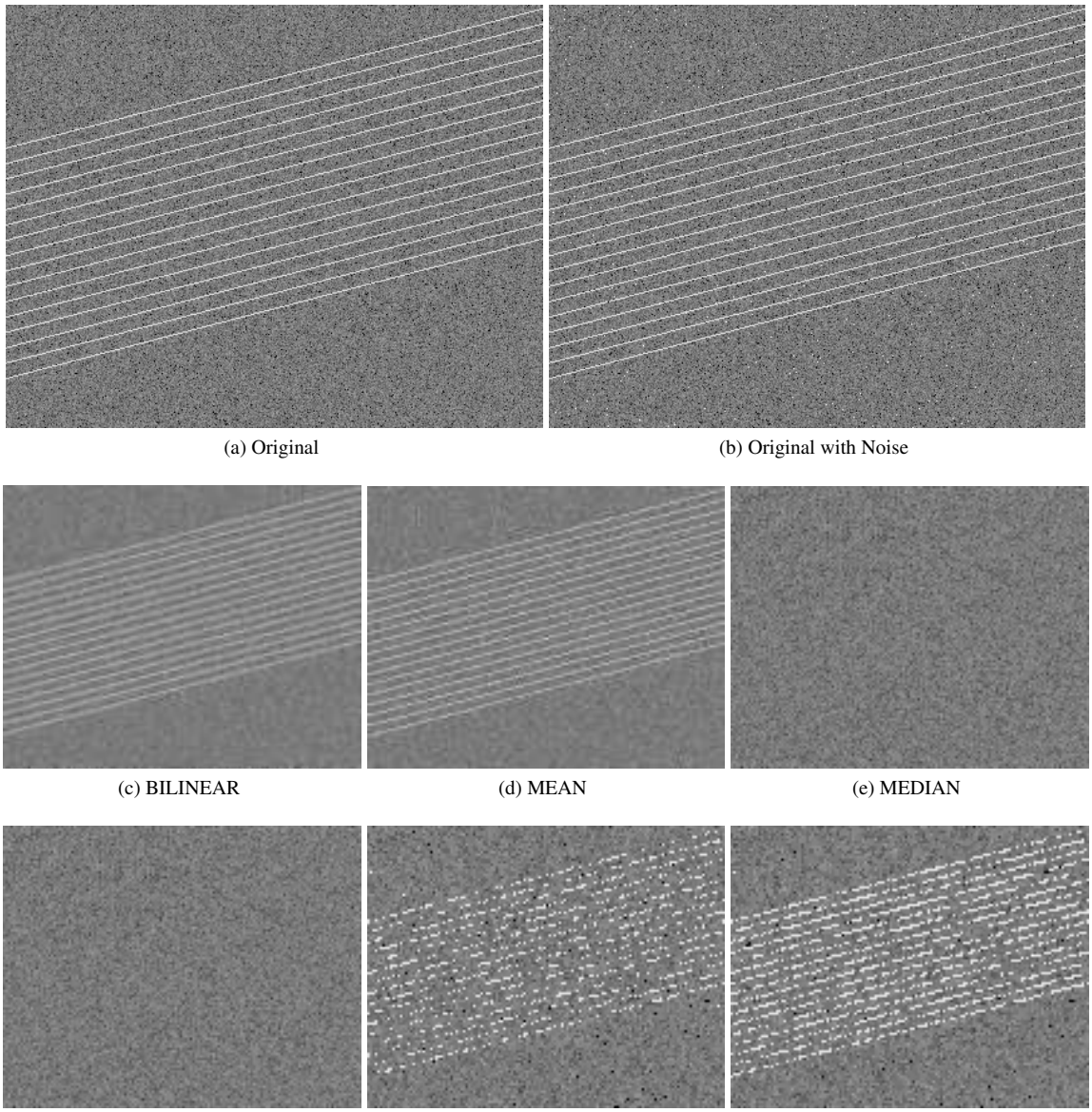
Query image	Clean	Salt and Pepper		Gaussian - variance		Speckle - variance	
		1%	5%	0.0025	0.01	0.0025	0.09
Original size	61.185	27.66	15.34	13.56	10.81	**35.56**	10.84
Mean	38.97	29.23	20.8	**27.91**	**22.39**	32.16	**23.31**
Median	43.4	36.47	16.8	17.48	12.16	22.66	11.46
Bicubic	41.12	28.83	20.09	25.82	19.86	**33.51**	22.09
F-transform	39.05	29.2	13.65	**26.86**	21.04	31.62	22.2
Pmode	**50.78**	**50.64**	**43.73**	18.69	13.27	32.88	12.16
Cmode	**50.7**	**50.69**	38.72	18.98	13.77	**33.23**	12.48

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2879473, IEEE Access

TABLE 5: Log-average Miss Rate (%) with test images, both clean and noisy, original and reduced by the indicated method. The noise level is indicated by the percentage (S&P) and variance (Gaussian and Speckle). The best precision is indicated in boldface.

Test image	Clean	Salt and Pepper		Gaussian - variance		Speckle - variance		Mixed	Mixed	Mixed
		1%	5%	0.0025	0.01	0.0025	0.09	(0.0001, 1%)	(0.0025, 1%)	(0.0001, 5%)
Original size	16.83	20.42	43.7	24.52	39.38	16.92	36.83	19.66	29.79	42.69
Mean	17.85	**19.18**	46.84	**18.72**	**32.67**	17.2	**27.63**	**19.67**	27.17	48.86
Median	17.91	27.3	82.97	62.24	83.95	21.97	82.97	31.37	72.42	84.25
Bicubic	16.14	21.18	50.65	21.6	40.36	**16.72**	32.89	22.65	29.65	52.11
F-transform	17.42	20.91	44.99	20.31	34.54	17.08	29.09	20.51	**26.22**	46.53
Pmode	20.28	20.28	**23.27**	43.65	69.71	21.55	74.63	21.25	46.54	**28.06**
Cmode	20.46	**19.83**	25.93	39.54	68.17	21.08	71.05	20.97	43.75	30.93

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2879473, IEEE Access



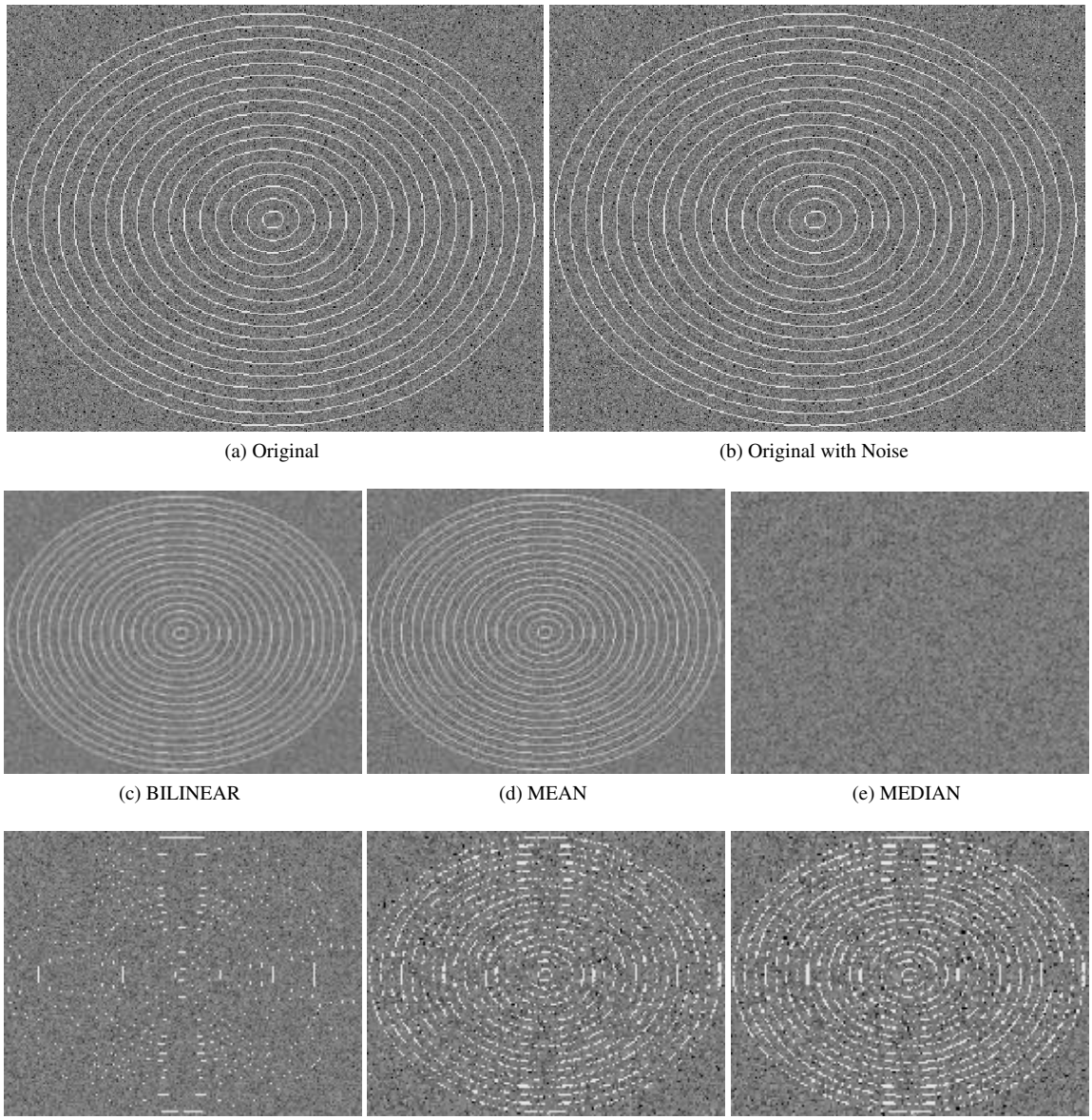
(a) Original (b) Original with Noise

(c) BILINEAR (d) MEAN (e) MEDIAN

(f) SHORTH (g) PMODE (h) CMODE

FIGURE 2: Lines image before and after reduction (not to scale).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2879473, IEEE Access



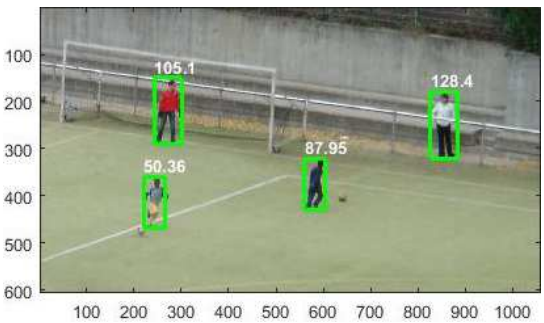
(a) Original (b) Original with Noise

(c) BILINEAR (d) MEAN (e) MEDIAN

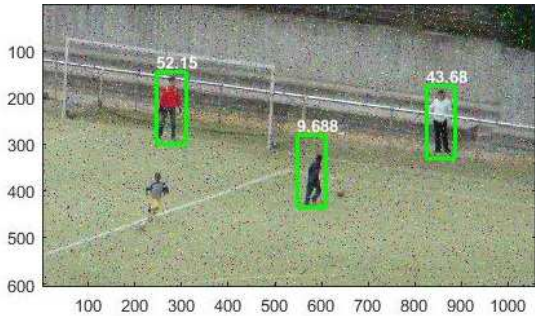
(f) SHORTH (g) PMODE (h) CMODE

FIGURE 3: Circles image before and after reduction (not to scale).

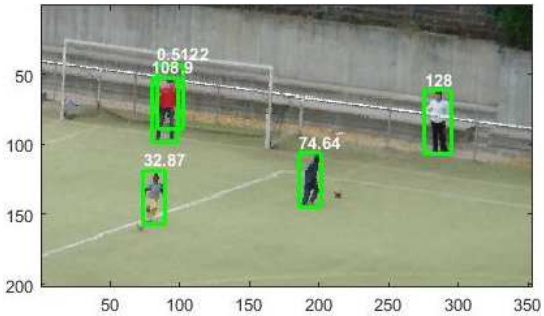
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2879473, IEEE Access



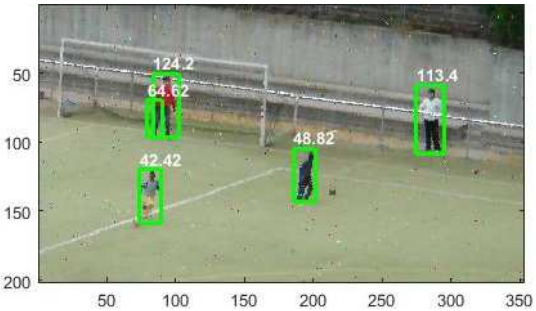
(a) Clean & Original



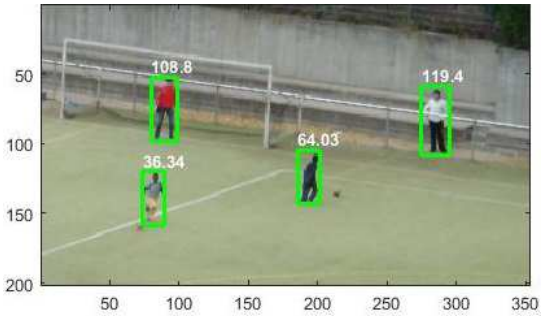
(b) Noisy & Original



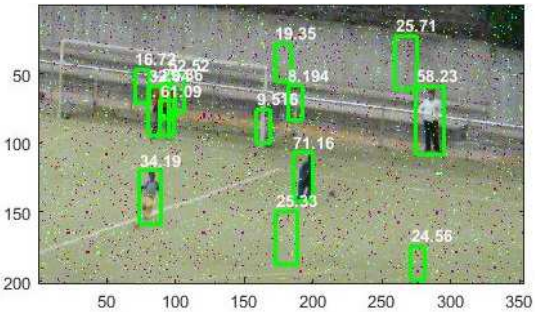
(c) Clean & Pmode-reduced



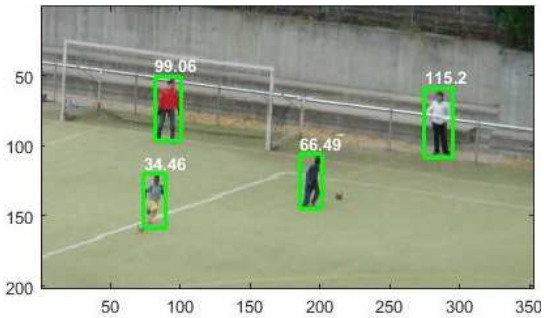
(d) Noisy & Pmode-reduced



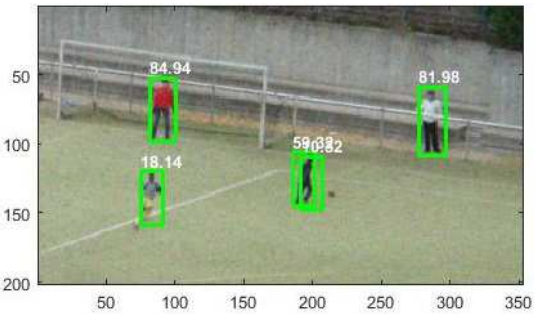
(e) Clean & Median-reduced



(f) Noisy & Median-reduced



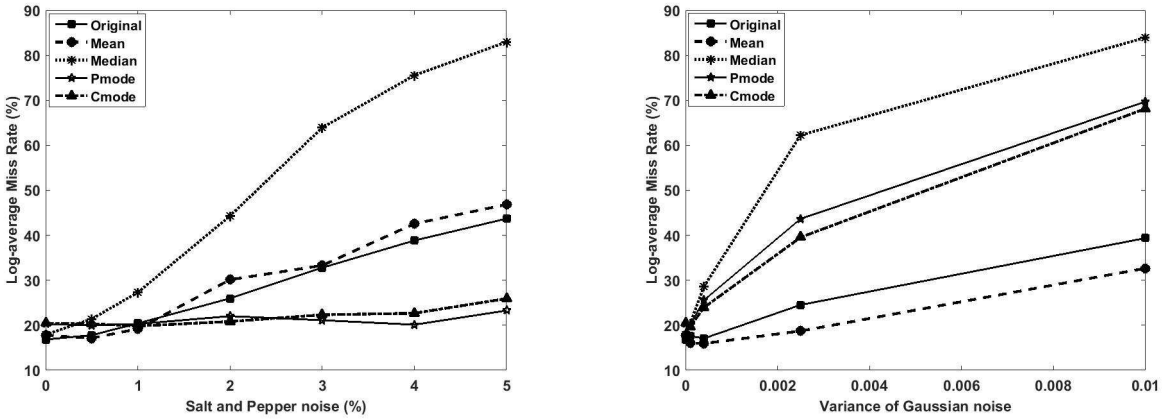
(g) Clean & Mean-reduced



(h) Noisy & Mean-reduced

FIGURE 4: Performance of detector with both original and reduced test images. Noisy images are corrupted with 5% Salt and Pepper noise and then reduced.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2879473, IEEE Access



(a) Salt and pepper noise

(b) Gaussian noise

FIGURE 5: Performance of the pedestrian detector with both original and reduced test images. ROC curves are obtained using Piotr's Matlab Toolbox [19], and with INRIA dataset.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2879473, IEEE Access

GLEB BELIAKOV (M'08–SM'08) received a PhD degree in Physics and Mathematics in Moscow, Russia, in 1992. He is currently a Professor with the School of Information Technology at Deakin University. His research interests are in the areas of fuzzy systems, aggregation operators, multivariate approximation, global optimization and computational mathematics. He is the author of over 150 research papers and two monographs in the mentioned areas, and a number of software packages. He serves as an Associate Editor of IEEE Transactions on Fuzzy Systems, Fuzzy Sets and Systems and TOP journals.

GITA DAS obtained her BTech and MTech degrees in Radio Physics and Electronics from the University of Calcutta, India, in 1990 and 1992 respectively, and PhD degree in Information Technology from Monash University, Australia in 2008. She has a broad exposure of working across projects including defence, medical, teaching and research. She has 15 research publications (including IEEE conferences). Her current work interests include data analysis, image analysis and retrieval, machine learning and deep learning.

HUY QUAN VU is a Lecturer in ICT at Central Queensland University (CQU), Australia. Prior to joining CQU, he was a Research Fellow in the Center for Applied Informatics at Victoria University. He received his Ph.D. from School of Information Technology, Deakin University. His research interests include data mining and social media analytics applications in tourism management.

TIM WILKIN (M'12) received the PhD degree in computer science from Deakin University, Burwood, Australia, in 2014. He is currently an Associate Professor with the School of Information Technology, Deakin University. His research interests cover the mathematical formulation of problems in image processing and sensor analytics, with a focus on real-time problems arising in robotics and cyber-physical systems. He is the author of more than 25 research papers in the mentioned areas and several software packages, and co-winner of the Best Paper award at PRICAI'00. He serves as a reviewer for the journals IEEE Trans. on Fuzzy Systems and Information Sciences (Elsevier), and is the Publications Co-Chair for FUZZ-IEEE 2019.

YONG XIANG (SM'12) received the Ph.D. degree in Electrical and Electronic Engineering from The University of Melbourne, Australia. He is a Professor and the Director of the Artificial Intelligence and Image Processing Research Cluster, School of Information Technology, Deakin University, Australia. His research interests include information security and privacy, signal and image processing, data analytics and machine intelligence, and Internet of Things. He has published 2 monographs, over 100 refereed journal articles, and numerous conference papers in these areas. He is an Associate Editor of IEEE Signal Processing Letters and IEEE Access. He has served as Program Chair, TPC Chair, Symposium Chair, and Session Chair for a number of international conferences.