engineering proceedings

MDPI

Proceeding Paper

Adaptive Gaussian and Double Thresholding for Contour Detection and Character Recognition of Two-Dimensional Area Using Computer Vision [†]

Nehal Abdul Rehman * **and Farah Haroon**

Department of Industrial Electronics Engineering, Institute of Industrial Electronics Engineering, Karachi 75300, Pakistan; farah@iiee.edu.pk

* Correspondence: nehalrehman.30@iiee.edu.pk

[†] Presented at the 2nd International Conference on Emerging Trends in Electronic and Telecommunication Engineering, Karachi, Pakistan, 15–16 March 2023.

Abstract: Contour detection with good accuracy is challenging in various computer-aided measurement applications. This paper evaluates the performance and comparison of thresholding and edge detection techniques for contour measurement along with character detection and recognition between images of high and low quality. Thresholding is one of the key techniques for pre-processing in computer vision. Adaptive Gaussian Thresholding (AGT) is applied to distinguish the foreground and background of an image, and Canny edge detection (CED) is used for spotting a wide range of edges. Adaptive Gaussian Thresholding works on a small set of neighboring pixels, while Canny Edge Detection takes high- and low-intensity pixels in the form of thresholds that are tested to find accurate contour measurements while retaining the maximum data contained within them. The results show that Adaptive Gaussian Thresholding outperforms Canny edge detection for both brightened sharp and blurry dull images.

Keywords: Adaptive Gaussian Thresholding; binarization; Canny edge detection; contour measurement; computer vision; image segmentation; Character Recognition

check for updates

Citation: Rehman, N.A.; Haroon, F. Adaptive Gaussian and Double Thresholding for Contour Detection and Character Recognition of Two-Dimensional Area Using Computer Vision. *Eng. Proc.* **2023**, *32*, 23. https://doi.org/10.3390/engproc2023032023

Academic Editors: Muhammad Faizan Shirazi, Saba Javed, Sundus Ali and Muhammad Imran Aslam

Published: 22 May 2023

1. Introduction

Computer vision is a field of study that focuses on how computers can be made to interpret and understand visual information, such as images and videos. Image segmentation is a specific task within computer vision that involves dividing an image into multiple segments or regions, each of which corresponds to a different object or part of the scene. This can be accomplished using a variety of techniques, such as thresholding, edge detection and more. Image segmentation based on contours and measurement to a close accuracy is one of the challenges in computer-aided projects currently. Contouring is used to identify the object in an image that can be used to analyze or observe something. Contours can help in counting and detecting humans, fruits, vegetables, animals, microorganisms and more and hence, therefore, are utilized in every sector where computer aid is needed, be it in detecting or measuring Bau-Kul(fruit) [1], white blood cells [2], road cracks [3], steel defects [4] and more.

It all starts with a pre-processing stage where with image binarization. Binarization is a technique for converting the pixel image into a binary image. Image binarization separates the pixel values into two groups. The foreground pixels are specified as black, and the background pixels are specified as white [5]. Adaptive Gaussian Thresholding works on images with inconsistent light or pixel intensities. It takes an odd integer as the block size, and thus every block can have a different thresholding value depending on its inconsistencies. The value of constant C is taken to subtract from the mean or weighted mean from all the regions (blocks) in order to return a desirable outcome [6,7]. Canny edge detection is a multi-stage algorithm used to detect the edge in an Image. Canny algorithms have been commonly used in scientific research and practical engineering projects among the various edge detection algorithms because of their outstanding performance. The threshold selection is an important factor that determines the output of the Canny algorithm. There are two fixed global threshold values to clear out the false edges in the conventional Canny edge detection algorithm [8,9].

A contour is an outline around something with clearly defined boundaries, which suggests that a computer is capable of computing gradient differences (significant variations in pixel values), determining whether the variations continue to exist, creating recognizable shapes, and then drawing a boundary around those shapes [10,11].

Image segmentation is one of the crucial problems in today's computer vision tasks. Edge and boundary detection can be applied for segmenting the boundary of a tumor from normal brain tissue [12] and for road boundary detection for smart driving systems and management and more.

The paper is organized with Section 1 discussing the importance of computer vision with a focus on contouring techniques. Section 2 provides the functionality of detection of 2D surfaces with retention of internal details. Results and discussions are presented in Section 3, and finally, Section 4 concludes the results and highlights the future avenues in medical imaging.

2. Materials and Methods

The whole operation of the ID Card detection system is depicted in the accompanying flowchart (Figure 1). Theoretically, the ID Card detection method employs both low-level and high-level processing. A mobile camera was used to take a high-quality image, while a low-quality image was captured using a pi-camera for low-level processing, as can be seen in Figure 2. The obtained RGB images were converted to grayscale photos in order to extract the Thresholding images.

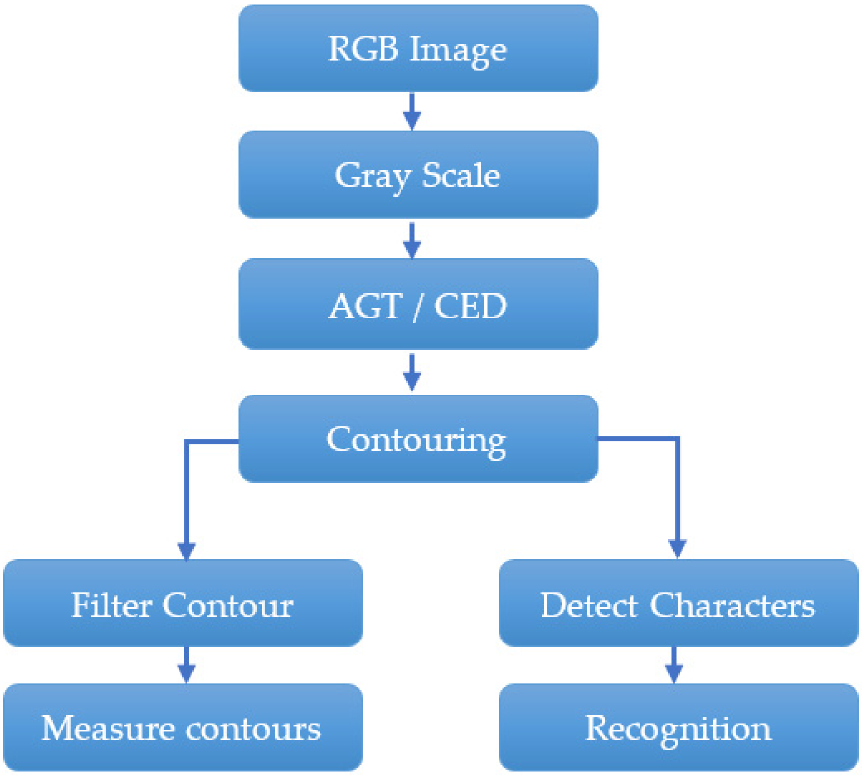
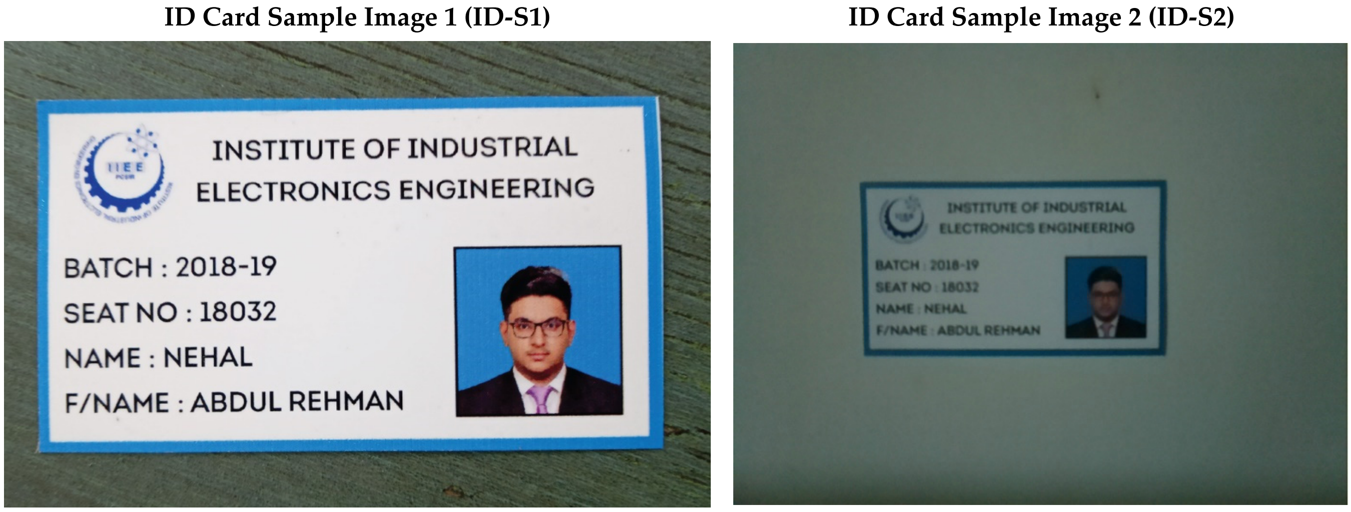


Figure 1. Flow diagram of surface detection (contour measurement and OCR).



ID Card Sample Image 1 (ID-S1)

INSTITUTE OF INDUSTRIAL ELECTRONICS ENGINEERING

BATCH : 2018-19

SEAT NO : 18032

NAME : NEHAL

F/NAME : ABDUL REHMAN

Sample size pixels: 1125 × 741—300 dpi

ID Card Sample Image 2 (ID-S2)

INSTITUTE OF INDUSTRIAL ELECTRONICS ENGINEERING

BATCH : 2018-19

SEAT NO : 18032

NAME : NEHAL

F/NAME : ABDUL REHMAN

Sample size pixels: 640 × 480—96 dpi

Figure 2. Image samples based on size and dpi.

To create the parametric shapes of ID Card samples using the Canny edge function, high-level processing was then applied to the image. Using the Thresholding and Canny Edge images, the next stage was obtaining the proper fits of contours around the sample images.

The implementation of algorithms for the contour fitting of ID cards, therefore, consists of the following consecutive steps: (a) image acquisition and pre-processing, (b) thresholding and canny edge detection (c) contouring and (d) text reading. These steps and their mathematical models have been described as following system architecture (Table 1):

Table 1. System architecture.

Algorithm	Actions
Start	
Step-1:	Read image into the Python-OpenCV then convert into grayscale image [13]
Step-2:	AGT and CED are applied for binary conversion
Step-3:	Find and measure contours for closest accuracy
Step-4:	OCR to read text from the binary image
Stop	

2.1. *Image Acquisition and Pre-Processing*

The images of different quality in terms of clarity and brightness are compared, and high-level processing is discussed in other steps in detail. The captured images were then saved as a .jpg and/or .png format for further processing.

In pre-processing, after the image is read in Integrated Development Environment (IDE), it is converted into grayscale for the algorithms to work and to binarize the images.

2.2. *Adaptive Gaussian Thresholding and Canny Edge Detection*

The sample images were segmented for AGT based on the mean thresholding value, which is based on the block size (a pixel with neighboring pixels) and a fine-tuned constant C. The action of thresholding the image produces binary images from a grayscale image. The AGT is used as an operation that involves tests against a function of threshold T of the form:

$$T(x, y) = m(x, y) - k * s(x, y) \tag{1}$$

where T is the threshold value at pixels x and y, m is the local mean of pixels in the block, s is the local standard deviation, and k is the user-defined constant that controls the sensitivity of the thresholding process.

In addition, Canny edge detection is an image segmentation technique that extracts useful structural information [14] from different vision objects and dramatically reduces the amount of data to be processed. This method is quite complex and has a five-step process [15] that:

(i) de-noises the image with a 5 × 5 Gaussian filter (default for Sobel operator):

$$g(m,n) = G_\sigma(m,n) * f(m,n), \text{where } G_\sigma = \frac{1}{\sqrt{2\pi\sigma^2}}\exp(\frac{-m^2+n^2}{2\sigma^2}) \tag{2}$$

(ii) calculates edge gradients and direction for each pixel:

$$G(m,n) = \sqrt{g_m^2(m,n) + g_n^2(m,n)}, \text{and } \theta(m,n) = \tan^{-1}[\frac{g_n(m,n)}{g_m(m,n)}] \tag{3}$$

(iii) applies non-maximum suppression (NMS) on edges obtained to thin out the edge ridges, and

(iv) sets a double threshold on all the detected edges to eliminate false positives.

It also analyzes all the edges and their connections to each other to retain the real edges and discard the weaker ones [14]. Here, the default Sobel kernel is used, which is 3 × 3 for the changes in the horizontal and vertical directions and is a combination of Gaussian smoothing and differentiation. The edge pixels are in an image, and there is no particular requirement that the pixels representing an edge are all contiguous [1].

2.3. Contour and Perimeters

In order to find a boundary (closed shape) of an object appearing in the foreground, contours are found using RETR_TREE, which is a contour retrieval mode and creates a perfect hierarchy to detect each contour and drawing contour using CHAIN_APPROX SIMPLE, which creates circles at the corners/vertices of drawn contours instead of lines to save memory, and, finally, contours are drawn by locating and visualizing in ID-S1 and ID-S2. This will evaluate the accuracy of the manually determined perimeter with computed ones. Perimeters are measured here as the vertical and horizontal lines drawn from the mid-point since the samples appear to be rectangular.

2.4. OCR Text Reader

Simultaneously, to compare the effect of AGT and CED on internal information of each technique performance, characters were detected and recognized. Finally, characters containing both English alphabets and numerals were retrieved to evaluate the functioning of AGT and CED.

3. Results and Discussion

Gray images produced from originals were fed raw to apply thresholding. For the vertical and horizontal perimeters, Adaptive Gaussian Thresholding (AGT) produced the most accurate results for ID-S1 at 23 as the block size and 2 as the constant value (C). Similarly, AGT was applied on ID-S2 to be as accurate as possible at 17 as the block size and 5 as the constant value (C) measurement. More internal data was visible at this block size and C, as can be seen in Figure 3.

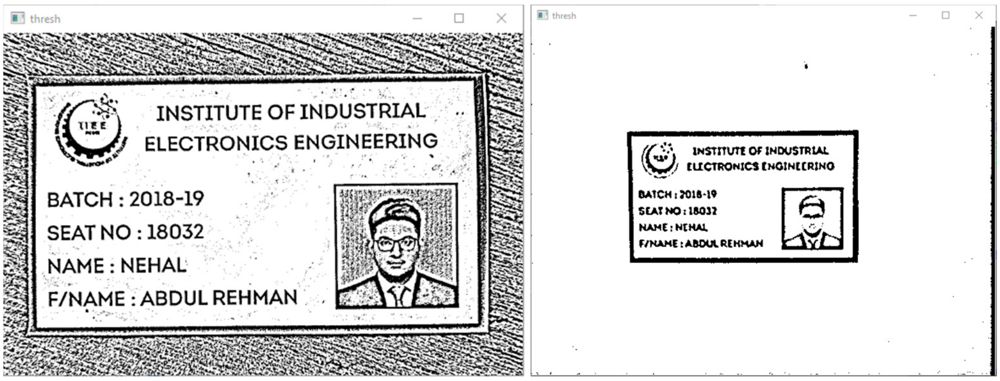


Figure 3. AGT applied to ID-S1 and ID-S2.

From histograms, the thresholding values for ID-S1 for upper and lower bounds were taken at 103 and 204, respectively. For ID-S2, the upper and lower bounds of 56 and 103, respectively, were considered for Canny edge detection (CED) because of the count of pixels with variance around ±30% for particular intensity and contour localization, resulting in better character detection of both samples; see Figures 4 and 5.

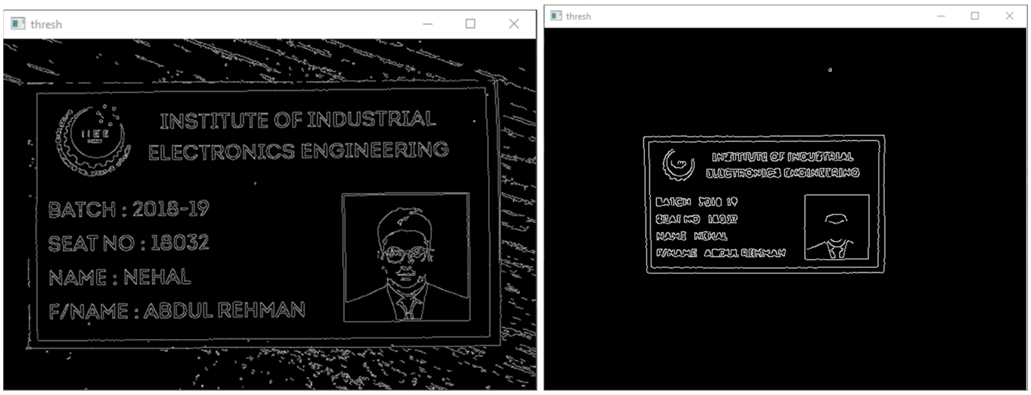


Figure 4. CED applied ID-S1 and ID-S2.

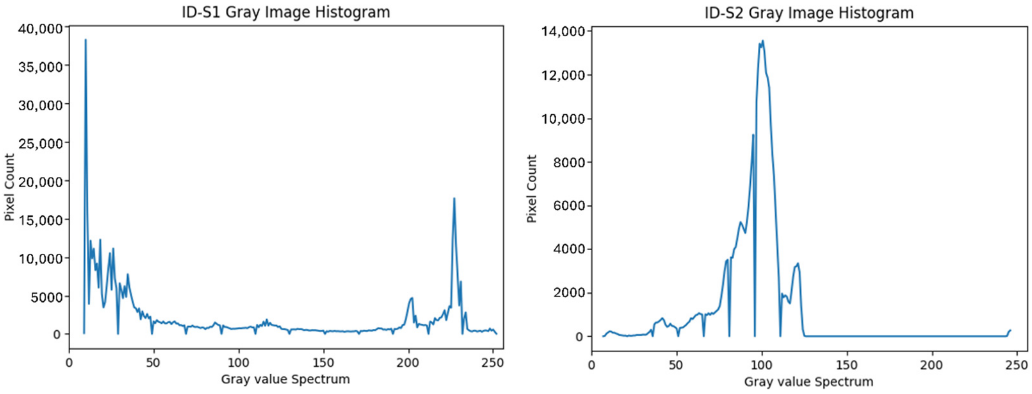


Figure 5. Histograms of grayscale ID-S1 and ID-S2.

We can see from Table 2 that ID-S1 is likely to be more suitable for producing stable and consistent results throughout the spectrum of AGT, and the block size can be considered larger than that of ID-S2 as far as the contour edges are concerned; however, CED for ID-S1, because of an inconsistent and edgy background, was not able to form a proper edge in order to draw a contour over it and is hence less accurate for this particular case. Conversion from pixels to inches with respect to DPI was performed, and it can be clearly seen that

AGT is more accurate overall for the perimeter measurements from both algorithms, having 98.71% and 99.12%, respectively. However, CED achieves 49.62% and 99.19% for ID-S1 and ID-S2, respectively. It took less than two seconds in computational time for each algorithm per image for processing.

Table 2. Contour detection and measurement accuracy.

Contour Technique	Perimeter Measurement		Overall Accuracy %	
	ID-S1	**ID-S2**	**ID-S1**	**ID-S2**
AGT	3.30 × 1.85, 3.19 × 1.76	3.33 × 1.93, 3.19 × 1.75	98.71	99.12
CED	Not Found, 3.19 × 1.75	3.33 × 1.91, 3.19 × 1.76	49.62	99.19

Later, internal content in the form of text and numerals was extracted to check the effect of both techniques on the pixel intensity within the contours, and characters were generally more detectable because of the noise. We found that the characters extracted from ID-S1, because of its crisp and sharp quality, with AGT achieved 91.86% and with CED 97.93% similarity to the original characters after passing through a smoothing filter. However, the result from ID-S2 AGT was more accurate, having 76.76% compared with CED, which achieves 34.73% as the overall accuracy for character extraction and recognition. Overall, a dataset of 40 images was considered for the average accuracies present in Tables 2 and 3.

Table 3. Character extraction (detection) and recognition accuracy.

Recognition Technique	Characters Extracted		Recognition Accuracy %	
	ID-S1	**ID-S2**	**ID-S1**	**ID-S2**
AGT	113	102	91.86	76.76
CED	98	94	97.93	34.73

4. Conclusions

In this paper, we compared AGT and CED algorithms on two different types of images to test the validity of their functioning, and from our results, we can conclude that Adaptive Gaussian Thresholding works better and outperforms Canny Edge Detection for both of the images, as it not only retains the edges but also the information residing inside the contours, because of the block size advantage over CED. We can also deduce that AGT block size is larger for images with better readability and quality; hence, less time is taken for the algorithm to work when compared to images with less readable or blurry content, which smaller blocks and more time. With this research, we can comprehend that AGT will be a better choice in industries such as medical science for localization, classification and segmentation of the malicious or mild visible symptoms of life-threatening diseases such as cancer because of drastic changes in pixel intensity. Whereas CED can be utilized where construction lines forming edges are needed to be extracted, such as for fingerprints and cracks, etc.

Author Contributions: Conceptualization, N.A.R.; methodology, F.H.; writing—original draft preparation, N.A.R. and F.H.; writing—review and editing, N.A.R.; supervision, F.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rahman, M.M.; Oliver, M.M.H. Detection and contouring of bau-kul using image processing techniques. *Ann. Bangladesh Agric.* **2019**, *23*, 15–25. [CrossRef]
2. Mandyartha, E.P.; Anggraeny, F.T.; Muttaqin, F.; Akbar, F.A. Global and adaptive thresholding technique for white blood cell image segmentation. *J. Phys. Conf. Ser.* **2020**, *1569*, 022054. [CrossRef]
3. Othman, Z.; Zukfily, S.N.A.; Kasmin, F.; Ahmad, S.S.S. Determination of Threshold Value in Canny Algorithm for Road Crack Detection. Presented at the 12th Malaysian Technical Universities Conference on Engineering and Technology (MUCET), Ayer Keroh, Malaysia, 16–18 November 2021.
4. Wang, S.; Wu, S.; Wang, X.; Li, Z. A Canny operator road edge detection method based on color features. *J. Phys. Conf. Ser.* **2020**, *1629*, 012018. [CrossRef]
5. Ingle, P.D.; Kaur, P. Adaptive thresholding to robust image binarization for degraded document images. In Proceedings of the 2017 1st International Conference on Intelligent Systems and Information Management (ICISIM), Aurangabad, India, 5–6 October 2017; IEEE: New York, NY, USA; pp. 189–193.
6. Gatos, B.; Pratikakis, I.; Perantonis, S.J. Adaptive degraded document image binarization. *Pattern Recognit.* **2006**, *39*, 317–327. [CrossRef]
7. Neogi, N.; Mohanta, D.K.; Dutta, P.K. Defect detection of steel surfaces with global adaptive percentile thresholding of gradient image. *J. Inst. Eng. Ser. B* **2017**, *98*, 557–565. [CrossRef]
8. Chen, Q.; Sun, Q.S.; Heng, P.A.; Xia, D.S. A double-threshold image binarization method based on edge detector. *Pattern Recognit.* **2008**, *41*, 1254–1267. [CrossRef]
9. Wang, Y.; Li, J. An improved Canny algorithm with adaptive threshold selection. *MATEC Web Conf.* **2015**, *22*, 01017. [CrossRef]
10. Seo, J.; Chae, S.; Shim, J.; Kim, D.; Cheong, C.; Han, T.D. Fast contour-tracing algorithm based on a pixel-following method for image sensors. *Sensors* **2016**, *16*, 353. [CrossRef] [PubMed]
11. Liang, L.; Lu, Y.; Zhu, H.; Ye, Z.; Chai, Z. Research on Contour Feature Extraction Algorithms for Arc-shaped Groove of Safety Seat Base. *J. Phys. Conf. Ser.* **2021**, *1828*, 012018. [CrossRef]
12. Zhu, Y.; Yan, Z. Computerized tumor boundary detection using a Hopfield neural network. *IEEE Trans. Med. Imaging* **1997**, *16*, 55–67. [PubMed]
13. Howse, J.; Minichino, J. *Learning OpenCV 4 Computer Vision with Python 3: Get to Grips with Tools, Techniques, and Algorithms for Computer Vision and Machine Learning*; Packt Publishing: Birmingham, UK, 2020; pp. 40–80.
14. Muthukrishnan, R.; Radha, M. Edge detection techniques for image segmentation. *Int. J. Comput. Sci. Inf. Technol.* **2011**, *3*, 259. [CrossRef]
15. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Analys. Mach. Intel.* **1986**, *8*, 679–698. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.