

Article

An Improved Median Filter Based on YOLOv5 Applied to Electrochemiluminescence Image Denoising

Jun Yang¹, Junyang Chen¹ , Jun Li^{1,*}, Shijie Dai² and Yihui He¹¹ College of Information Engineering, Sichuan Agricultural University, Ya'an 625000, China² College of Science, Sichuan Agricultural University, Ya'an 625000, China

* Correspondence: lijun@sicau.edu.cn; Tel.: +86-130-5659-1398

Abstract: In many experiments, the electrochemiluminescence images captured by smartphones often have a lot of noise, which makes it difficult for researchers to accurately analyze the light spot information from the captured images. Therefore, it is very important to remove the noise in the image. In this paper, a Center-Adaptive Median Filter (CAMF) based on YOLOv5 is proposed. Unlike other traditional filtering algorithms, CAMF can adjust its size in real-time according to the current pixel position, the center and the boundary frame of each light spot, and the distance between them. This gives CAMF both a strong noise reduction ability and light spot detail protection ability. In our experiment, the evaluation scores of CAMF for the three indicators Peak Signal-to-Noise Ratio (PSNR), Image Enhancement Factor (IEF), and Structural Similarity (SSIM) were 40.47 dB, 613.28 and 0.939, respectively. The results show that CAMF is superior to other filtering algorithms in noise reduction and light spot protection.

Keywords: denoise; YOLOv5; electrochemical reactions; ECL image; light spot



Citation: Yang, J.; Chen, J.; Li, J.; Dai, S.; He, Y. An Improved Median Filter Based on YOLOv5 Applied to Electrochemiluminescence Image Denoising. *Electronics* **2023**, *12*, 1544. <https://doi.org/10.3390/electronics12071544>

Academic Editors: Francesco Beritelli and Javid Taheri

Received: 15 January 2023

Revised: 14 March 2023

Accepted: 22 March 2023

Published: 24 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Electrochemiluminescence is a phenomenon of excited state luminescence formed by electron transfer reactions between electrical biomass near the working electrode. ECL has many advantages. For example, ECL can be changed to control the potential it exerts and ensure no background light interference, and it is also easy to manipulate. As a result of these characteristics, the advantages of ECL include increased controllability, quick response, high sensitivity, and simplicity. Therefore, ECL has been widely used in analytical applications, including medical diagnosis [1–3], chromatography [4,5], food [6–8], water testing [9–11], bioanalytical detection [12–14], and the analysis of various species [15–17]. ECL has also been used to monitor enzyme reactions [18]. However, when carrying out ECL research, researchers first need to capture the ECL image, and then research and analyze its light spot. With the trend of simple and intelligent devices and the rapid development of smartphones, today's smartphones have powerful computing abilities, analytical capabilities, and cameras, so they can be used for ECL visualization analysis. Based on these advantages, the smartphone system with the ability to analyze ECL images has gradually become a commonly used device to capture ECL images. At present, some novel, portable, visual, and efficient methods based on mobile terminals have been used to capture and detect ECL images [19,20]. However, in these ECL images captured by mobile phones, there are many unavoidable noises in the whole image in addition to the light spots, and these noises contain a large amount of interference information, which affects the accuracy of subsequent image analysis. Therefore, it is meaningful to develop a method to remove noise to improve the accuracy of the analysis. However, the location of and quantity of noise and even noise pixel values are not fixed, and the light spots in some ECL images are many and small, even not far from the size of the noise. Therefore, choosing an appropriate denoising algorithm is the key to removing the noise and improving the subsequent image analysis.

In the field of image-denoising algorithms, the Median Filter (MF) [21–23] is one of the commonly used denoising algorithms. The main working function is replacing the current pixel value with the median value calculated from the pixel values in the neighborhood of the current pixel to allow the pixel value of the current pixel to be close to the surrounding pixels, thereby eliminating the isolated noise pixels. MF works well when the noise density is not very high. However, when the probability of noise is relatively high, the denoising effect of the image processed by MF is poor, leaving a lot of unprocessed residual noise [24,25]. To remove the noise completely, it is necessary to increase the size of MF, but this will cause the image to become blurred. The ECL image belongs to the image with a high probability of such noise, and the noise is mainly distributed around and inside the light spot. Therefore, processing with a smaller size of MF will lead to the phenomenon of more residual noise while processing with a larger size of MF will lead to edge cutting and blurring of the light spot, and even light spot weakening and light spot disappearance.

To solve such problems, an Adaptive Median Filter (AMF) [26–28] was proposed. According to preset conditions, AMF dynamically changes the window size of the median filter until the conditions are met. Therefore, it takes into account both noise reduction and detail protection effects. However, AMF and the Modified Decision-Based Unsymmetric Trimmed Median Filter (MDBUTMF) [29,30], Noise Adaptive Fuzzy Switching Median Filter (NAFSMF) [31–33], Different Applied Median Filter (DAMF) [34], The Adaptive Iterative Fuzzy Filter (AIFF) [35,36], and Probabilistic Decision-Based Filter (PDBF) [37] have the same operation mechanism: firstly, whether the current pixel is noise is detected. If so, the new value calculated by the respective calculation methods is used to replace the current pixel value. However, this is their weakness. They can only play their role in dealing with salt-and-pepper noise (SPN) because they are all filtering algorithms improved from MF to better deal with SPN. Additionally, there are only two kinds of pixel values of SPN, 0 or 255, which, respectively, represent the maximum or minimum pixel value. Moreover, SPN is evenly distributed in the image. For SPN, they can directly determine whether the current pixel is noise. However, the pixel value of the noise in the ECL image is not fixed, and the distribution is not uniform, mostly distributed around and inside the light spot.

Therefore, the above seven filtering algorithms cannot determine whether the current pixel is noise in ECL images, so they are not very good and correct to adjust the size of the current filter in real-time. Therefore, the above improved algorithm based on MF cannot be applied to ECL images. In order to make MF applicable to noise reduction in ECL images, MF must solve a problem, which is how and when to adjust the size of the filter to take into account both noise reduction and detail protection. For ECL images, the light spot is the only object in the image to be protected, and the large quantity of impurities lying around and inside the light spot represents the noise that we should remove. Moreover, the larger the size of a filter, the better the noise reduction and the worse the detail protection, and vice versa. Therefore, on the basis of MF, we propose a Center-Adaptive Median Filter (CAMF). The size of the CAMF can be adjusted according to the center and boundary of the light spot and the position of the current pixel. When the current pixel is far away from the light spot boundary, the filter size is larger, and vice versa. This ensures that the CAMF removes as much noise around and within the light spot as possible while protecting the edge of the light spot.

However, our CAMF cannot automatically obtain the center and boundary information of the light spots. In order to obtain the center and boundary of light spots in the ECL image, You Only Look Once (YOLO) [38] is used. As a model of object detection, YOLO can find specific objects in a given image and obtain their positions, so we can use YOLO to obtain the boundaries and centers of the objects. Therefore, in this paper, we first use YOLO to obtain the boundaries and centers of all the ECL images. Based on this boundary and center information, we can use our CAMF to denoise the ECL images. However, it is important to note that YOLO is only an auxiliary algorithm for CAMF, which does not

have noise reduction ability; it simply helps CAMF to obtain the position information of the light spot in the image before denoising the image.

Compared with other object detection algorithms, YOLO performs well in multi-scale object detection. As an excellent object detection model, YOLO has a fast detection speed and good detection accuracy. In many cases, it can even detect items in real-time. Overall, YOLO is a model that combines speed and accuracy. In the subsequent section, we describe how CAMF works.

Before beginning the narrative of the subsequent sections, we first describe the structure of our article. Section 2 describes the related work of YOLO and the Median Filter. Section 3 describes the dataset we used and our proposed method. Section 4 describes the experimental results and discussions. Section 5 concludes the paper.

2. Related Work

2.1. YOLO

2.1.1. Development of the YOLO Algorithm

With the rapid development of deep learning, designing deep convolutional neural networks for object detection has become a popular task, and has driven considerable achievements in the field of computer vision [39]. However, it is difficult to deploy in mobile device scenarios. You Only Look Once version 5 (YOLOv5) is used in this work to complete the frame detection of light spots in the image and realize the pre-processing operation of the data before the CAMF. From the initial You Only Look Once version 1 (YOLOv1) [38], the idea of single-stage object detection was proposed for the first time, and end-to-end object detection was realized. In 2018, You Only Look Once version 3 (YOLOv3) [40] was designed with two main innovations, one is the use of a unique residual structure Darknet-53, and the other is the adoption of the Feature Pyramid Networks (FPN) architecture suitable for solving multi-scale detection problems in the network. The YOLOv5 model proposed later paid more attention to the training strategy of the model, such as Mosaic data enhancement, which adopted the random extraction of four sample images and the Mosaic combination of four sample images through random scaling, random cropping, and random arrangement. Since each training sample has its corresponding label frame, a new sample image is obtained after splicing the four images. For details, please refer to Figure 1. This operation can greatly enrich the sample types that can be learned by the previous network, improve the accuracy and robustness of the model, and contribute to the loss convergence of the network. Inspired by the work of Reproducible Visual Geometry Group (RepVGG) [41], You Only Look Once version 7 (YOLOv7) [42] was proposed in August 2022. It introduced the idea of retransmission into the YOLO algorithm and implemented the dynamic label allocation strategy, mainly dealing with the distribution of output results from different branches of the tail of the YOLOv7 network.

2.1.2. Practical Application of YOLO

In recent years, the YOLO algorithm has received a lot of attention due to its lightweight, concise, and efficient performance and has been applied by many scholars in biology, chemistry, and agriculture. Natheer Khasawneh [43] et al. implemented a high-precision real-time K-complex detection in the form of migration learning using YOLOv3 as a model architecture. Luqman Ali [44] et al. developed a laboratory safety monitoring system for detecting the behavior of laboratory personnel, which mainly utilized the algorithm of YOLOv5. Marco Sozzi [45] et al. collected a dataset of grapefruit images under different natural conditions, compared the performance of the algorithms of the YOLO family on this dataset, and concluded that the YOLOv5x model is practical for this problem conclusion. However, although the above scholars achieved good results on their own research problems, the research process is only a simple application of the algorithm and lacks further reflection in the YOLO model; in addition, the proposed model is not well adapted to other areas of research.

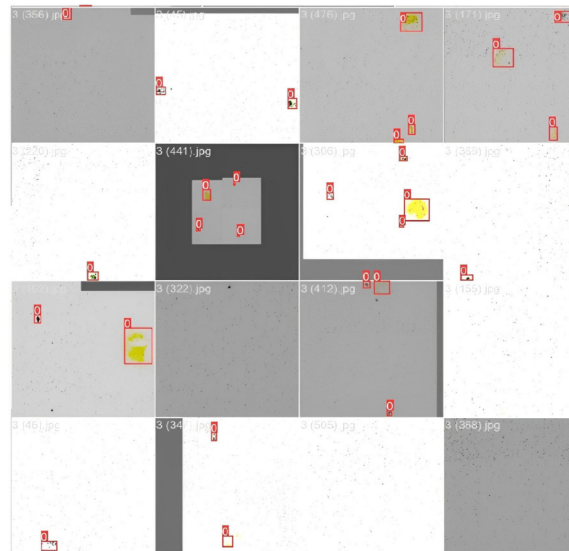


Figure 1. Mosaic data enhancement.

In developing the YOLO algorithm, the first thing to consider is the model architecture. Yanzhao Yang [46], based on the problem that the YOLO algorithm cannot collect features better for small-sized targets, added one upsampling in the neck of the network, thus completing the construction of the small target detection layer, which significantly improved the feature extraction capability of the YOLO algorithm. To overcome the high hardware dependence of the model on embedded deployment, Miaolong Cao [47] et al. fused the GhostNet model with YOLO, which effectively shortened the number of parameters of the detection model and further improved the detection speed without degrading the accuracy. Kun Chen [48] et al. combined the Cycle-GAN method with the YOLO algorithm and enhanced the samples by Cycle-GAN to generate a new dataset, which improved the robustness of the YOLO algorithm. In this paper, we propose a novel denoising method, while the YOLO algorithm is used for auxiliary detection to achieve efficient detection of electrochemical images.

2.2. Median Filter

MF is a nonlinear filtering algorithm that replaces the current pixel value with the median value calculated from the pixel values in the neighborhood of the current pixel. In this paper, we assume that the input image is x with size $M \times N$, and that $x_{i,j}$ represents the pixel in the image with pixel position (i, j) , where $(i, j) \in A \equiv \{1, \dots, M\} \times \{1, \dots, N\}$. Below, we present a brief review of the MF filter; using $x_{i,j}(W)$ to represent a window of size $(2W + 1) \times (2W + 1)$ centered at (i, j) , $x_{i,j}(W)$ is defined as:

$$x_{i,j}(W) = \{(a, b) : |a - i| \leq W, |b - j| \leq W, (a, b) \in A\} \quad (1)$$

We define $x_{i,j}^{med}(W)$ as the median value of all pixel values in $x_{i,j}(W)$ sorted in ascending order. Therefore, the working principle of MF is defined as:

$$x_{i,j} = x_{i,j}^{med}(W) \quad (2)$$

Here, $(i, j) \in A \equiv \{1, \dots, M\} \times \{1, \dots, N\}$.

The workflow of MF is shown in Figure 2.

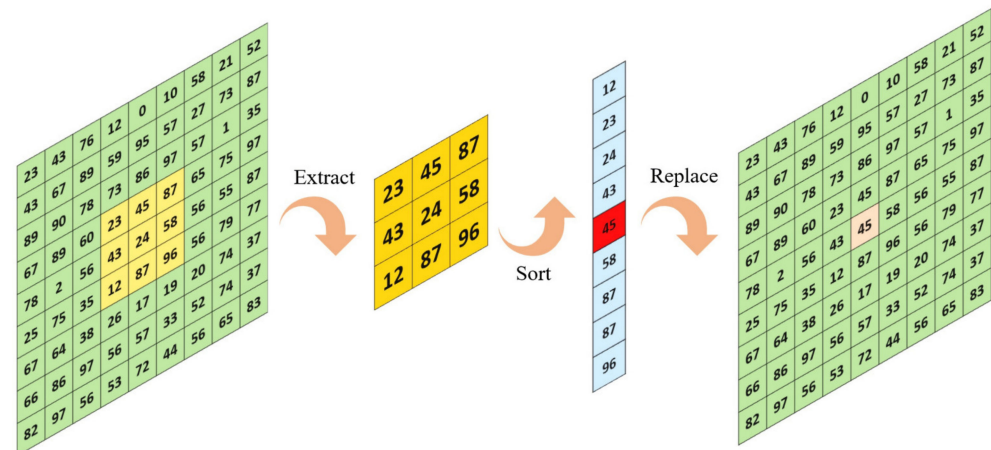


Figure 2. Median Filter.

The whole filtering process can be completed by traversing the filter through the whole image.

There are only three types of pixels in an ECL image: light spot, noise, and irrelevant pixels. In the ECL image, the pixel values of irrelevant pixels are mostly 0 or 255. Therefore, this mechanism of using the median value of the pixel values in the filter window instead of the current pixel can help MF to remove the noise surrounded by irrelevant pixels or light spots. The size of MF never changes from the beginning to the end of the image processing, so when the size of MF is small, it can effectively protect the edge, contour, and other details of the light spot. When the size of MF is large, although it can effectively remove the noise around and inside the light spot, it causes serious weakening of the edge and contour of the light spot, and even the phenomenon of light spot disappearance. To solve this problem, we proposed the CAMF based on the improved MF. The specific processing procedure of CAMF will be described in the next section.

3. Materials and Methods

3.1. Materials

All the ECL images used in our experiment were collected from the College of Science, Sichuan Agricultural University. Before we captured the ECL images, we first needed to pick a smartphone to use for subsequent photography. In order to select a suitable smartphone, we first used five different smartphones to capture the same light spot, then observed and compared the results. The results taken with these five smartphones are shown in Figure 3. The smartphones we used were the Xiaomi P30, OPPO Reno8 Pro+, Honor X20, iPhone 13, and Huawei P30. As we can see in Figure 3. The brightness levels of the light spots captured with the Xiaomi P30, OPPO Reno8 Pro+, and Honor X20 are low. The brightness of the spot captured by the iPhone 13 is good but not as bright as that captured by the Huawei P30. The stronger the brightness of the captured light spot is, the more beneficial it is for subsequent research. Therefore, we finally chose the Huawei P30 as the camera equipment for subsequent photography. The shooting parameters were ISO = 3200, S = 1/17 s, EV = 0, F = 2.2, and the focal length was 26 mm. We first took 600 images inside a self-made 3D-printed portable electroluminescence camera box and Huawei P30 smartphone. In order to enhance the robustness of our approach, we also captured 600 images outside the camera box, not limited to the region inside the camera box. All ECL images were shot in total darkness. Therefore, the images captured are almost all black, which is not conducive to the researchers' observation and subsequent research work. In order to solve this problem, the images should first be preprocessed. Since most areas of the original image are black and the pixel value of the black part is [0, 0, 0], we change the pixel value of the pixel points that meet the conditions to [255, 255, 255]. is the pixel value of white. Again, white is the maximum pixel value of any color. The effect of our image preprocessing is shown in Figure 4.

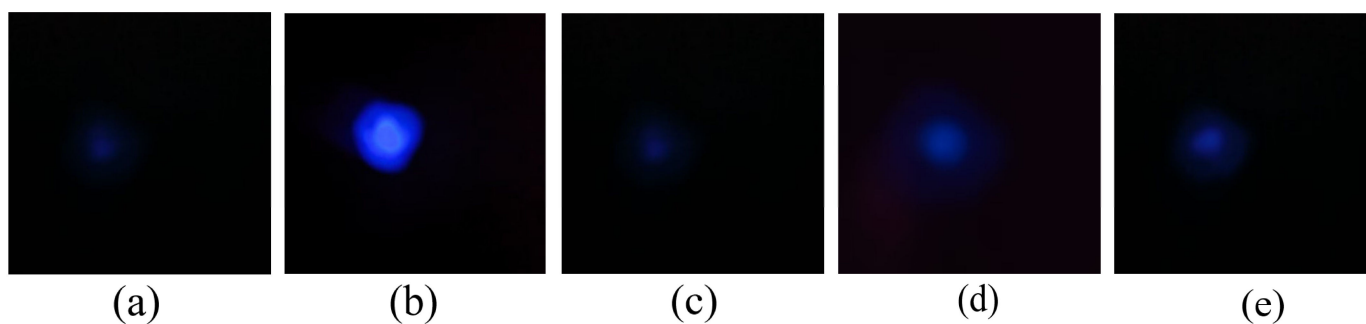


Figure 3. Comparison between five mobile phones. (a) Image captured by Xiaomi P30; (b) Image captured by Huawei P30; (c) Image captured by OPPO Reno8 Pro+; (d) Image captured by iPhone 13; (e) Image captured by Honor X20.

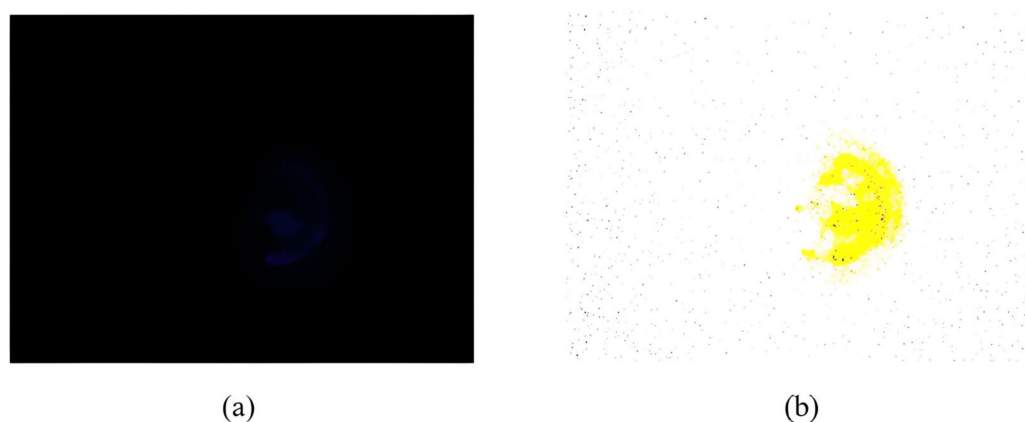


Figure 4. Image comparison before and after processing. (a) Original image; (b) Processed image.

After collecting the ECL images, we cropped the size of those images to 3840×3840 pixels, saved them in JPG format, and made them into our dataset. Some images from the dataset are shown in Figure 5. The dataset contains a total of 1200 images, of which 475 contain a single light spot, 329 contain two light spots, 262 contain three light spots, and 134 contain four or more light spots. In order to see the distribution relationship between the number of images and the number of spots in the dataset more intuitively, we drew the density plot formed by the number of included light spots and the number of images. The density plot is shown in Figure 6. For a density plot, the peaks of the density plot are at the locations where there is the highest concentration of points. As can be seen in Figure 6 the highest peak of the density plot is exactly when the number of light spots is one, which means that our dataset has the largest number of images with one light spot. We can also see that the peak of the density map is clustered at two, three, and four light spots. This means that our dataset consists of a large number of images with few light spots.

For the convenience of subsequent experiments, we adjusted the size of all images in the dataset to 3840×3840 . In the training of YOLOv5, we randomly divided the data set into the training set, verification set, and test set according to the ratio of 8:1:1.

3.2. Methods

3.2.1. YOLOv5

Before using CAMF to reduce the noise of the image, we first need to know the center and boundary information of the light spot in the image. In this paper, we first use YOLOv5 to detect the light spots in the image, so as to obtain some information about the light spots. In this paper, YOLO is a unity of speed and accuracy of the object detection model, which can effectively detect each light spot in ECL images and obtain its position information. In this work, version s of YOLOv5 is used, the version with the smallest network scale, which is an appropriate choice in the mobile device scenario for the pursuit of lightweight

performance. In this paper, YOLOv5s are compared with YOLOv3 and YOLOv7 in terms of various indicators. The results show that YOLOv5s has the best effect and can be effectively used to solve the light spot frame detection operation. The YOLOv5s network model contains four main components, namely, the input terminal, the backbone network, the neck, and the prediction layer. The model diagram is shown in Figure 7.

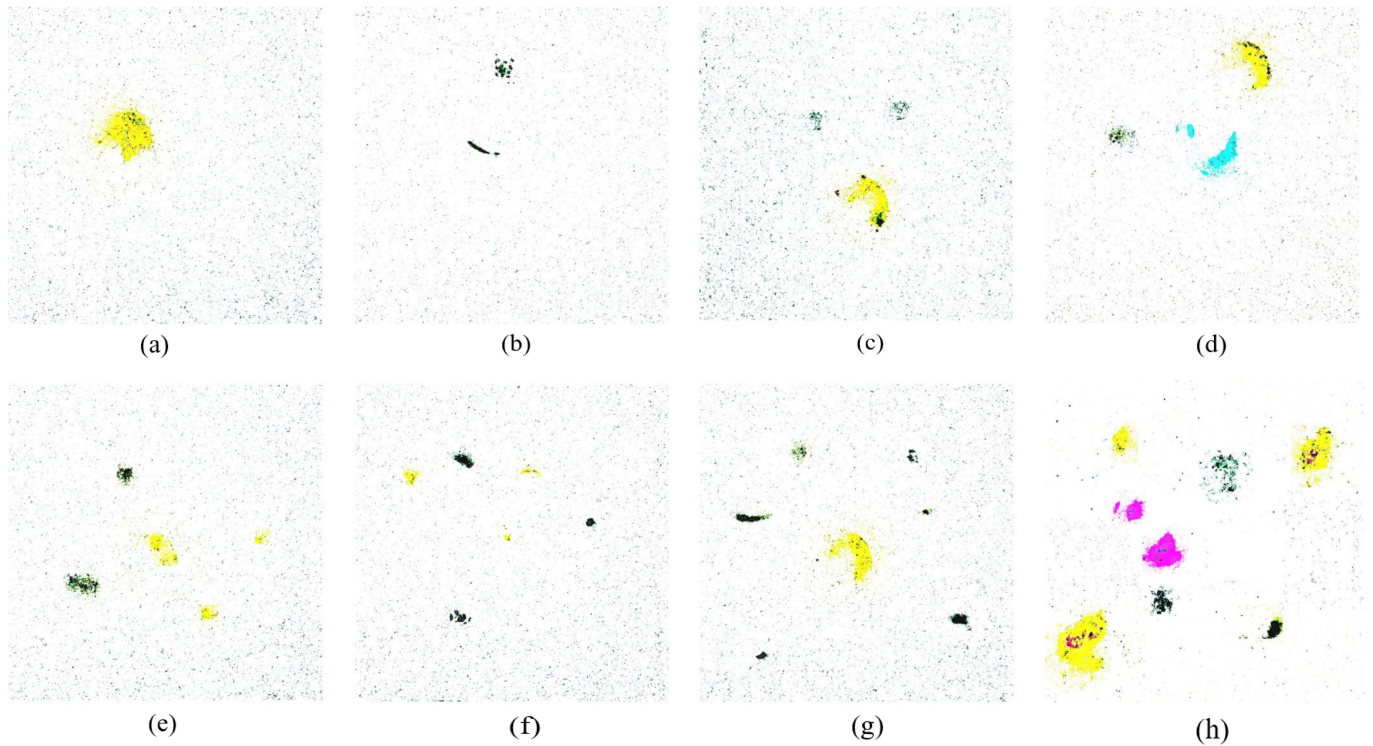


Figure 5. Data presentation. (a) Image with one light spot; (b) Image with two light spots; (c) Image with three light spots; (d) Image with four light spots; (e) Image with five light spots; (f) Image with six light spots; (g) Image with seven light spots; (h) Image with eight light spots.

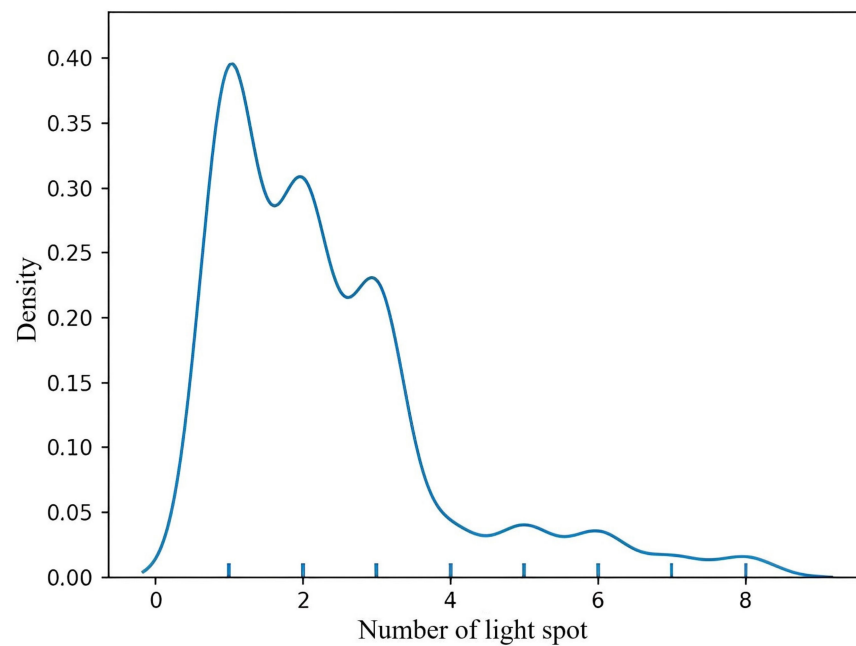


Figure 6. Density plot.

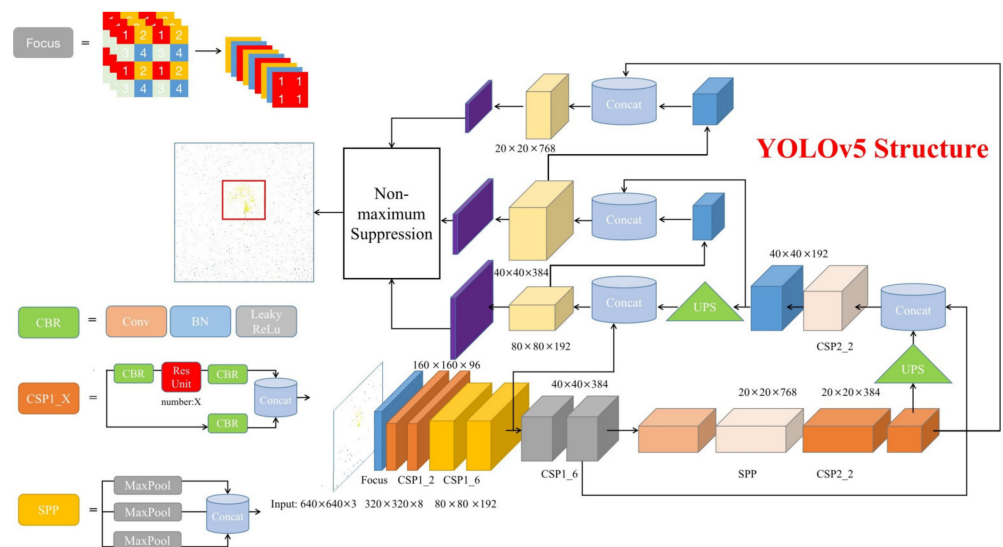


Figure 7. YOLOv5 model diagram.

In the experiments of this paper, the image at the input will be resized to 640×640 in order to minimize the loss of image information and reduce the computational cost of the network. In the process of image resizing, the network adaptively adds the least edge black pixel bars to the original image, so the original shapes of the images are not disturbed. At the same time, the input side also uses Mosaic data enhancement to expand the number of samples and accelerate the learning speed of the network. During the training phase, the YOLOv5s network continues to output the prediction frame, and reverse update and optimize the network parameters by calculating the gap loss between the prediction frame and the real frame.

The backbone network integrates the Focus and Cross-Stage-Partial (CSP) structures, with a total of 45 convolutional layers. The Focus structure is mainly used for slicing feature maps, which will improve the feature extraction ability of the subsequent convolution kernel. The CSP structure draws on the design idea of a Cross-Stage-Partial Network (CSPNet) [49] and is used in the backbone and Neck parts of the network, which are called $CSP1_x$ and $CSP2_x$, respectively. The operation purpose of CSP is to split the obtained feature map so that part of it is a convolutional operation and part of it is only connected with the results after the previous part of the convolution operation, which effectively reduces the computation amount.

In YOLOv5s, the so-called Neck module can be presented in the form of a Feature Pyramid Network (FPN) and is also composed of a Path Aggregation Network (PAN). This structure can sample input images multiple times from top to bottom, complete data fusion, and finally improve the feature extraction capability of the network.

How YOLOv5 detects the boundary of the light spot is shown in Figure 8. After the light spot boundaries of ECL images are detected by YOLOv5s, the center and boundary information of each light spot in each image can be obtained. Based on this information, we will use CAMF for image noise reduction.

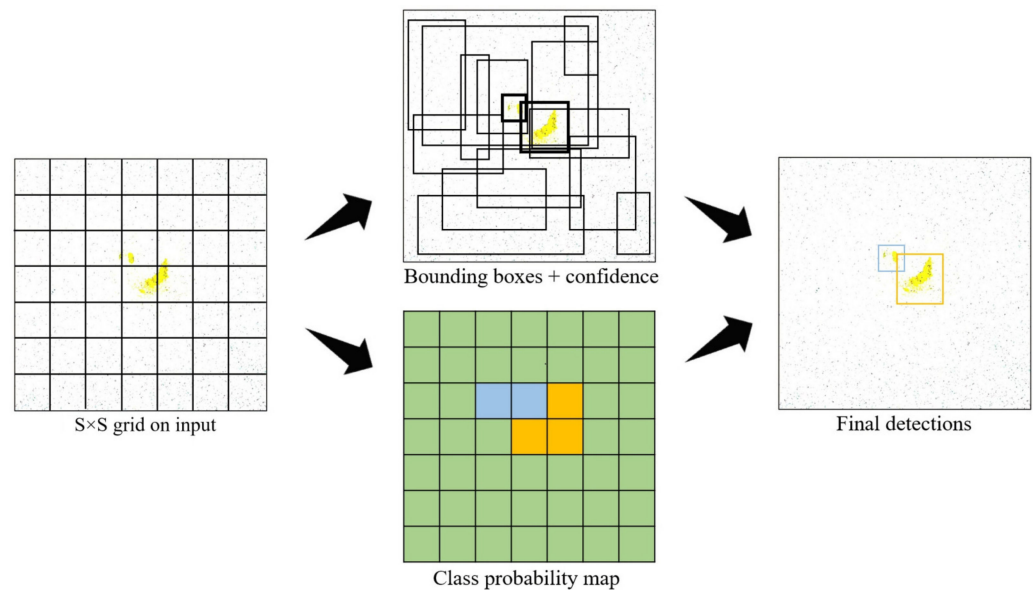


Figure 8. YOLOv5 was used to detect the light spot of the ECL images. In the Final detections, the orange and the blue frame are the boundary frames corresponding to the two light spots.

3.2.2. CAMF

After detecting the light spot in the image by YOLOv5, we can obtain the center and boundary information of the light spot. We assume that the obtained light spot center coordinate is (c_x, c_y) . We can also obtain information on the boundary frame of the light spot. We assume that the width of the boundary frame of the light spot is w and the height is h , and we assume the number of light spots is n . We also assume that the input image is x with size $M \times N$, and that $x_{i,j}$ represents the pixel in the image with pixel position (i, j) , where $(i, j) \in A \equiv \{1, \dots, M\} \times \{1, \dots, N\}$. Unlike the MF, we using the $x_{i,j}(W)$ to represent a window of size $W \times W$ centered at (i, j) .

Since the number of light spots in ECL images may be one or more, we start with the case of only one light spot. Let us assume that the current pixel is located at (i, j) . We hope that if the current pixel is outside the boundary frame of the light spot, the further away the current pixel is from the boundary frame, the larger the size of CAMF will be and the faster it will grow, so as to ensure that the noise outside the light spot can be eliminated as much as possible. The reason why the size of CAMF increases rapidly as the distance between the current pixel and the boundary frame increases is that when the current pixel is outside the boundary frame, we should think more about how to better reduce the noise and think less about how to protect the light spots. A larger size has a stronger denoising capability.

However, if the current pixel is inside the boundary frame of the light spot, we hope that the further away the current pixel is from the boundary frame, the larger the size of CAMF will be and the slower it will grow. Unlike the above case when the light spot is outside the boundary frame, in this case, we cannot let the size of CAMF increase rapidly as the current pixel moves away from the boundary frame because the rapid increase in size will lead to the destruction of the light spot where the current pixel is located. In this case, we should think more about how to better protect the light spot and think less about how to reduce the noise. So, the size of CAMF should increase slowly as the current pixel moves away from the boundary frame to prevent the interior of the light spot from being destroyed by the rapid increase in the size of CAMF.

When there are multiple light spots in the image, we just need to take the nearest light spot to the current pixel as the reference light spot. In this way, the multi-spot case can be transformed into a single-spot case for processing. The purpose of taking the nearest light spot as the reference light spot is to prevent, as much as possible, each spot from being damaged. This is because the closer the light spot is to the current pixel, the smaller the size of CAMF will be, and a smaller size has a better capacity for protecting light spots.

However, too large a size will not only lead to too long a processing time but may also lead to the destruction of the light spot; thus, we cannot increase the size of CAMF without boundaries. Therefore, we set an upper limit value S for the size of CAMF and hope the size is not larger than $S \times S$. In this paper, we take a quarter of the minimum of both the width and height of the smallest light spot as S , so S is defined as:

$$S = \frac{\min_{1 \leq k \leq n} \{ \min(h_k, w_k) \}}{4} \tag{3}$$

Here, h_k and w_k are the height and width of the k th light spot.

At the same time, we set a lower limit for the size of CAMF. We hope that the size of CAMF is not less than 3×3 , because too small a size cannot achieve the desired noise reduction effect.

Synthesizing the above discussion, we assume that the current size of CAMF is $W \times W$. If the current pixel is outside the boundary frame of the conference light spot, which is the nearest light spot, the size W is defined as:

$$W = \min \left(2 \left(e^{\lfloor d_{i,j}^{near}(x) / s_{i,j}^{near}(x) \rfloor - 1} - 1 \right) + 3, S \right) \tag{4}$$

where e is the natural constant, $\lfloor \cdot \rfloor$ is rounded down and $s_{i,j}^{near}(x)$ is the distance between the center of the nearest light spot and the boundary frame of the nearest light spot in the direction pointing to $x_{(i,j)}$. The graphical description can be seen in Figure 9. Therefore, $s_{i,j}^{near}(x)$ is defined as:

$$s_{i,j}^{near}(x) = \frac{(i - c_x^{near})^2 + w_{near}^2}{4(j - c_y^{near})^2} \tag{5}$$

where w_{near} is the width of the nearest light spot, (i, j) represents the coordinates of the current pixel, and (c_x^{near}, c_y^{near}) represents the coordinates of the center of the nearest light spot.

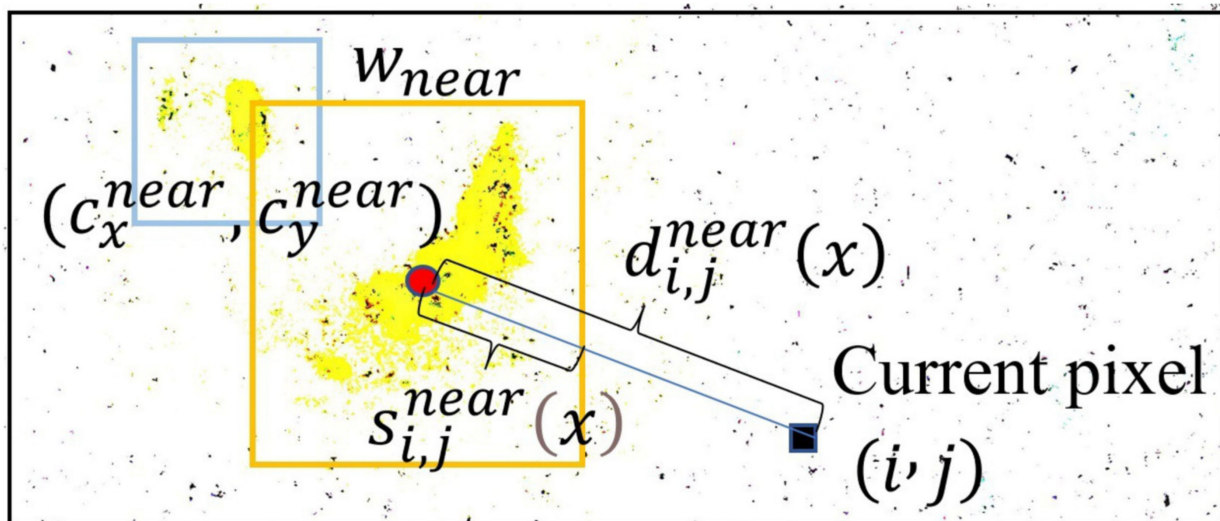


Figure 9. The graphical description of $s_{i,j}^{near}(x)$ and $d_{i,j}^{near}(x)$.

$d_{i,j}^{near}(x)$ is the distance between the current pixel and the center of the nearest light spot. In this paper, we use the Euclidean distance as the distance formula to measure the distance between two points. The graphical description is shown in Figure 9. Therefore, $d_{i,j}^{near}(x)$ is defined as:

$$d_{i,j}^{near}(x) = \sqrt{(i - c_x^{near})^2 + (j - c_y^{near})^2} \tag{6}$$

where (i, j) represents the coordinates of the current pixel and (c_x^{near}, c_y^{near}) represents the coordinates of the center of the nearest light spot.

If the current pixel is inside the boundary frame of the nearest light spot, the W is defined as:

$$W = \min\left(2\lfloor \log_{10}\left(\frac{s_{i,j}^{near}(x)}{d_{i,j}^{near}(x)}\right)\rfloor + 3, S\right) \tag{7}$$

where $\lfloor \cdot \rfloor$, $d_{i,j}^{near}(x)$ and $s_{i,j}^{near}(x)$ are defined as in Equation (4).

Unlike the MF, we assume $x_{i,j}(W)$ is a window of size $W \times W$ centered at (i, j) . Since W is already an odd number according to the above definition of W , we can just set the size of the window equal to the size of the CAMF. $x_{i,j}(W)$ is defined as:

$$x_{i,j}(W) = \left\{ (a, b) : |a - i| \leq \frac{W - 1}{2}, |b - j| \leq \frac{W - 1}{2}, (a, b) \in A \right\} \tag{8}$$

Similarly, we also define $x_{i,j}^{med}(W)$ as the median value of all pixel values in $x_{i,j}(W)$ sorted in ascending order. Therefore, the working principle of CAMF is defined as:

$$x_{i,j} = x_{i,j}^{med}(W) \tag{9}$$

Here $(i, j) \in A \equiv \{1, \dots, M\} \times \{1, \dots, N\}$.

Synthesizing the above corollaries, we next present the process of implementation of the proposed filter in Algorithm 1.

Algorithm 1. Center-Adaptive Median Filter (CAMF).

For each pixel $(i, j) \in A$ of the image x , do

- (1) Initialize $W = 3$
 - (2) Compute $s_{i,j}^{near}(x)$, $d_{i,j}^{near}(x)$, S and $S_{i,j}^{med}(W)$.
 - (3) If $s_{i,j}^{near}(x) < d_{i,j}^{near}(x)$, go to step (4); Otherwise, go to (5).
 - (4) $W = \min\left(2\left(e^{\lfloor d_{i,j}^{near}(x)/s_{i,j}^{near}(x)\rfloor - 1} - 1\right) + 3, S\right)$, go to (6).
 - (5) $W = \min\left(2\lfloor \log_{10}\left(\frac{s_{i,j}^{near}(x)}{d_{i,j}^{near}(x)}\right)\rfloor + 3, S\right)$, go to (6).
 - (6) $x_{i,j} = x_{i,j}^{med}(W)$, and stop.
-

Based on the above processing method, the ECL image noise reduction can be realized by traversing all the pixels of the image with CAMF.

Since the ECL images are color images with three channel layers, namely R, G, and B, it is only necessary to perform the above operations separately for each layer and then combine the three layers to form an image.

The operation flow of CAMF is shown in Figure 10.

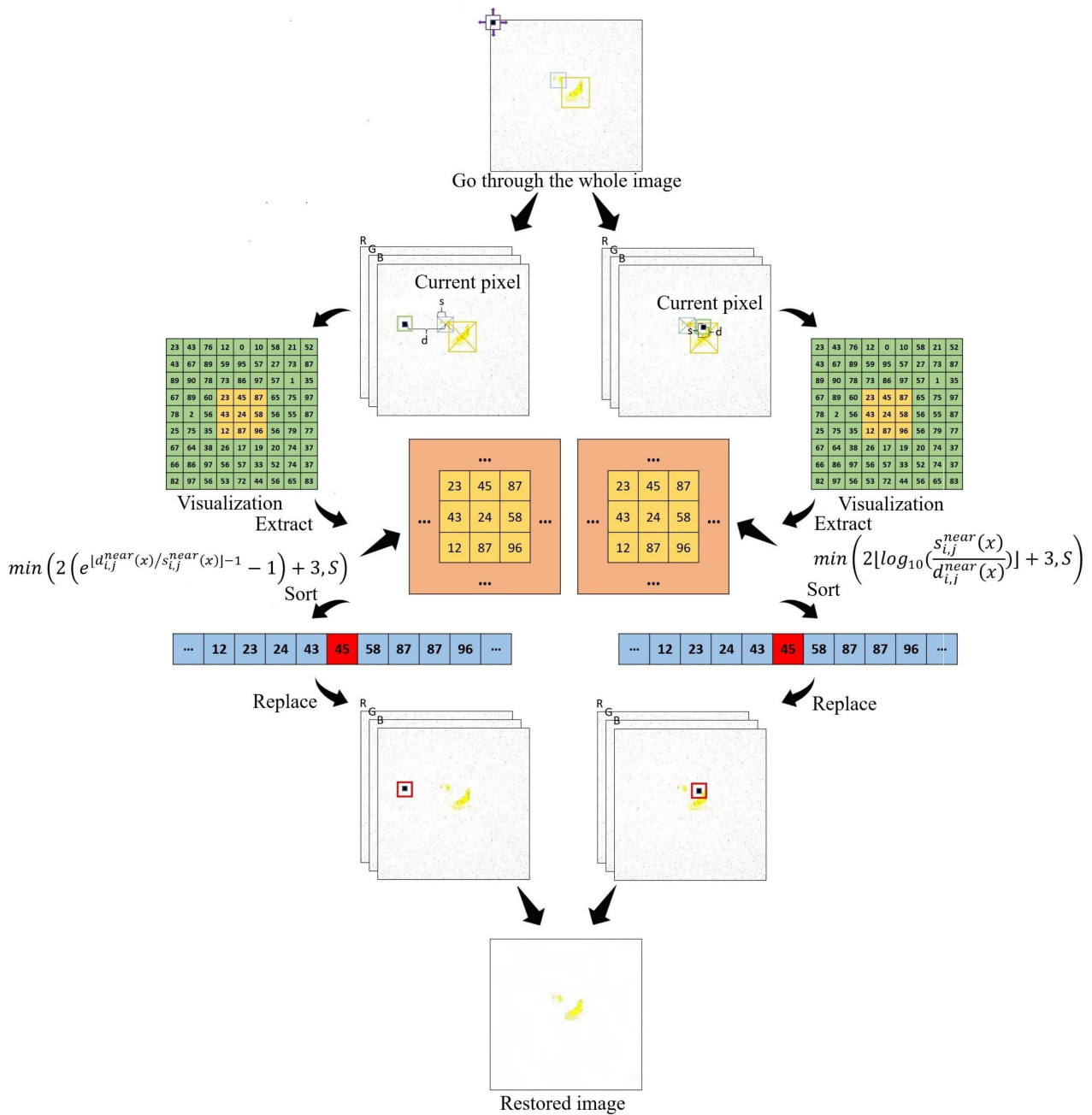


Figure 10. The process of CAMEF.

4. Results and Discussion

4.1. Comparison between YOLOv3, YOLOv5 and YOLOv7

In order to evaluate the detection performance of YOLOv5, YOLOv3, and YOLOv7 in terms of ECL images, this paper introduces four measurement indicators: Precision, Recall, F1-score and mean Average Precision (mAP) curve.

Precision is the probability of being correctly predicted, among all samples predicted as positive, which is defined as:

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

A recall is the probability that among the positive samples of the original sample, the final one is correctly predicted to be a positive sample, which is defined as:

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

The F1-score represents the harmonic average evaluation measure of precision and recall, which is defined as:

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (12)$$

In these formulas, FN is the number of mispredicted negative samples and FP is the number of mispredicted positive samples. TN is the number of correctly predicted negative samples and TP is the number of correctly predicted positive samples.

The vertical axis of the mAP curve represents the precision and the horizontal axis represents the recall. For a model, under a certain threshold, the model considers the samples whose results are greater than the threshold as positive samples, and the samples whose results are less than the threshold as negative samples. Thus, a point on the mAP curve represents the recall and precision corresponding to the threshold at this time. The whole mAP curve produces a transition from a high threshold to a low threshold. The mAP curve reflects the balance between recall and precision of the model. So, mAP is defined as:

$$mAP = \frac{\sum_{i=1}^k AP_i}{k} \quad (13)$$

AP_i is defined as:

$$AP_i = \sum_{i=1}^{n-1} (r_{i+1} - r_i) P_{inter}(r_i + 1) \quad (14)$$

where r_i is the i -th Recall from the Precision interpolation in ascending order, and $P_{inter}(r_i)$ represents the Precision interpolation from the i -th Recall. In brief, AP is the area under the accuracy and recall curves, and mAP is the average of the various AP s.

In this paper, we randomly select 1000 ECL images from the total dataset for comparative experiments, and use four measurement indicators, Precision, Recall, F1-score, and the mAP curve, to measure many aspects of the training and prediction results of YOLOv3, YOLOv5, and YOLOv7. The training comparison results of each model are shown in the following figures.

According to the results of Precision, as can be seen in Figure 11, the precision of the light spot frame detection of YOLOv3, YOLOv5, and YOLOv7 in the image reaches 86.9%, 98.8%, and 91.2%, respectively, among which the results of YOLOv5 meet the expectations of this paper. Through its high precision, the ECL images in the image can be efficiently identified to complete the frame detection and improve the next CAMF filtering algorithm processing.

In terms of the recall rate, as shown in Figure 12, the YOLOv5 model is 2.89% and 4.55% higher than the YOLOv3 and YOLOv7 models, respectively, with a value of 96.1%. A combination with the precision, that is, the F1-score, is shown in Figure 13. YOLOv5 achieves 98.1% performance, balancing the recall rate and accuracy, while the F1-scores of YOLOv3 and YOLOv7 are only 91.7% and 92.0%; the main reason is that the model size of YOLOv5 is suitable for this task. Too large a model is prone to the overfitting phenomenon, while too a small model will cause underfitting. Meanwhile, YOLOv5 combines the genetic algorithm with the K-means clustering method when calculating the anchor box, which is a unique technique in the YOLO algorithm series. In the training process of YOLOv3, if two objects' centers are a grid at the same time and assigned to the same anchor frame, the previous object is overwritten by the later object, which leads to the neglect of positive samples. Therefore, with the advanced clustering adopted in

YOLOv5, the genetic algorithm continuously adjusts the length and width of the anchor frame during the training process, which can accurately match the specific object size.

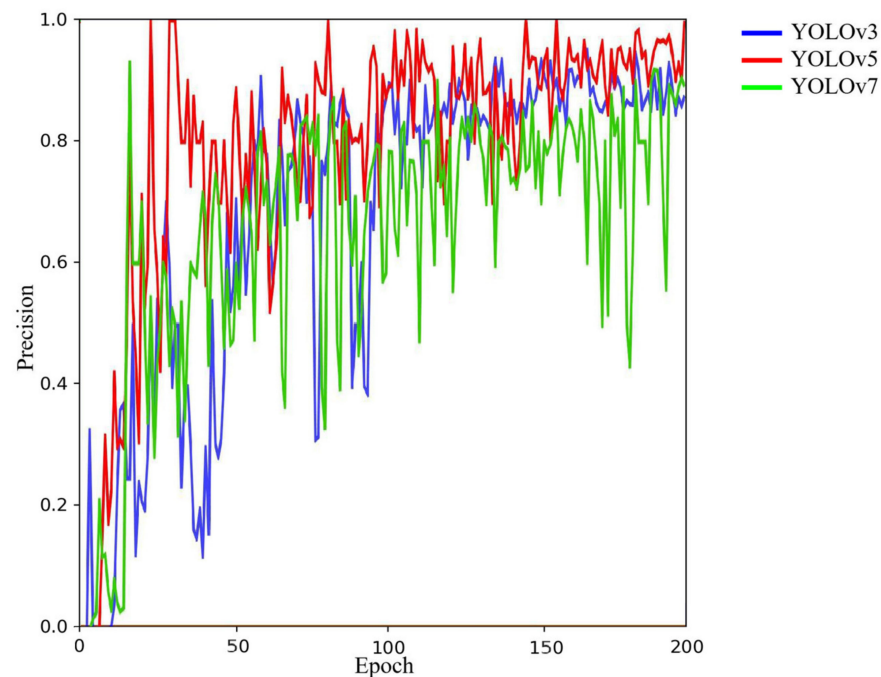


Figure 11. Precision curve.

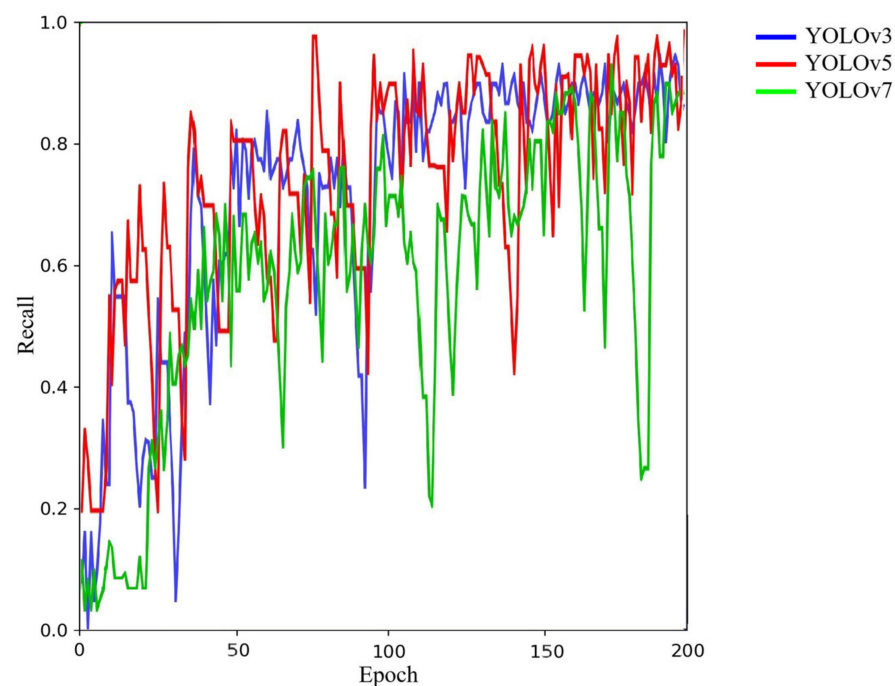


Figure 12. Recall curve.

As can be seen from Figures 14 and 15, the precision of the YOLOv5 algorithm adopted in this paper is far better than that of the YOLOv3 and YOLOv7 algorithms. When the confidence gradually increases, the precision of YOLOv5 is stable at a high level, indicating that most ECL images can be framed with high confidence, and the trained model has good robustness in this task. Compared with YOLOv7, the precision confidence curve of YOLOv7 is not as good as that of YOLOv5. The precision of YOLOv7 is close to 100% when the confidence threshold is around 0.7, while the precision of YOLOv5 is 100% when the

confidence threshold is 0.58. Due to the weak network fitting ability of the YOLOv3 model, it is difficult to show good precision performance under a high confidence threshold; a confidence threshold close to 0.85, a slump, indicates that it cannot be applied in this task.

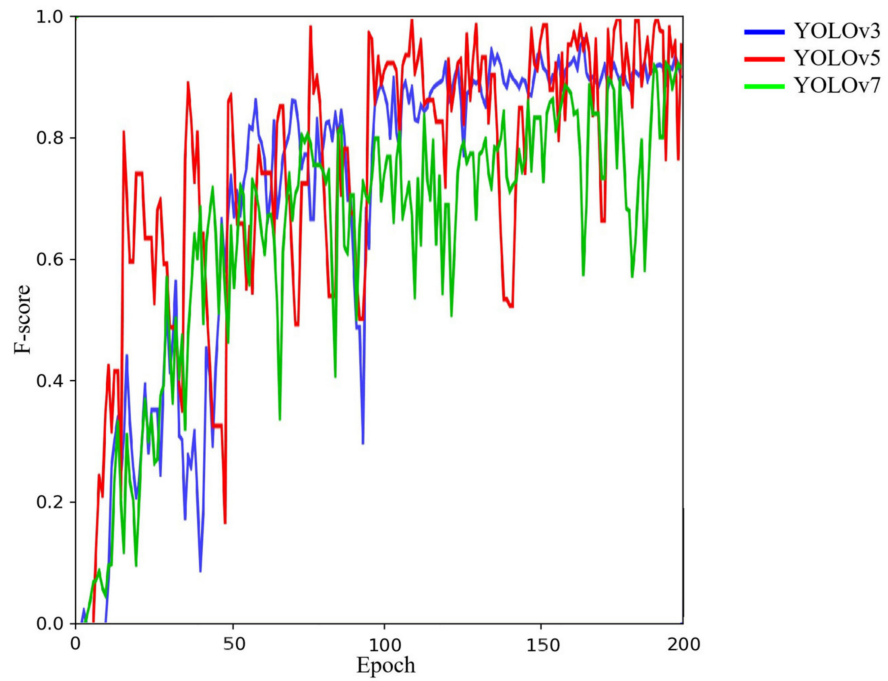


Figure 13. F1-score curve.

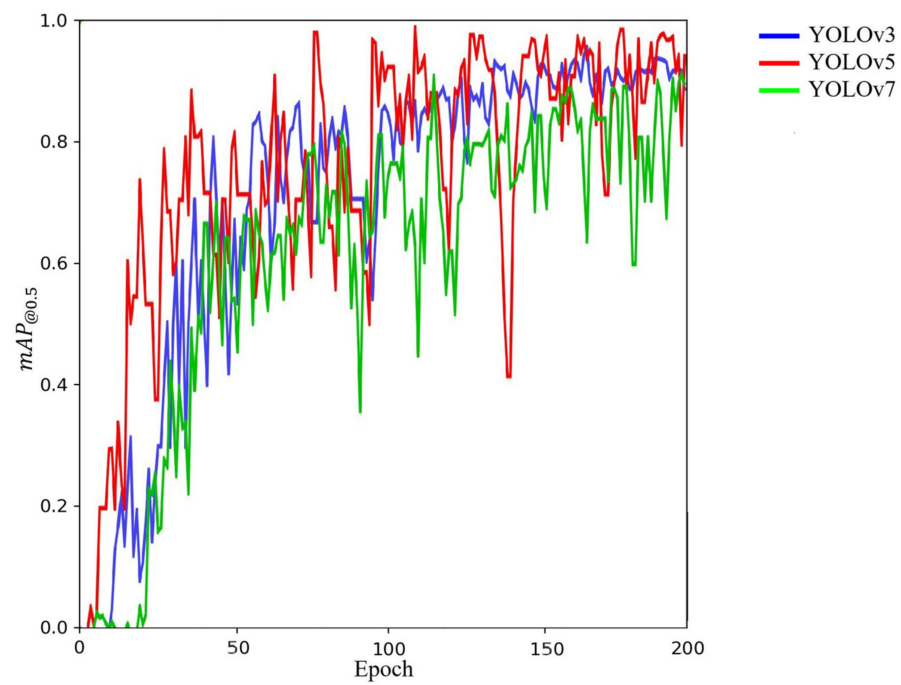


Figure 14. $mAP_{@0.5}$ curve.

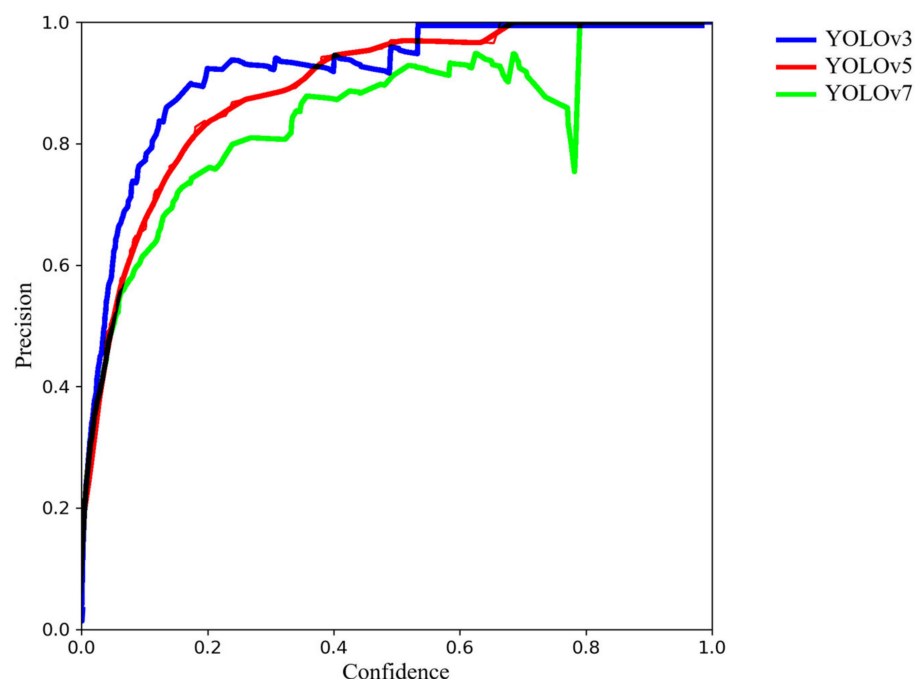


Figure 15. Precision-confidence curve.

As shown in Figure 16, we visualized the detection results of the three YOLO models and plotted the accuracy-recall curves, where the horizontal axis is the recall, the vertical axis is the accuracy, and the Average Precision (AP) is the area of the PR curve consisting of precision and recall with respect to the X/Y axis. Generally speaking, the larger the AP value, the higher the detection accuracy of the corresponding category. By adjusting the threshold of the intersection ratio, the samples with a probability greater than the threshold are positive cases, and all the later ones are negative cases; thus, multiple values of Precision and Recall can be obtained. In the figure, the Recall and Precision of the three models are inversely changed, i.e., one becomes higher and one indicator decreases accordingly, and the enclosed area becomes a very intuitive evaluation criterion. It can be seen that YOLOv5's model has a Precision value close to 1.0 in the interval where Recall lies between 0 and 0.78, while YOLOv3 is only able to maintain this performance in the interval where Recall lies between 0 and 0.2, and YOLOv7 decreases the Precision value earlier. Overall, YOLOv5 has a larger area under the Precision-Recall curve than YOLOv3 and YOLOv7, with a larger change from Recall values after 0.8. Although both YOLOv3 and YOLOv7 have these characteristics, the former has a single structure, lacking the Focus structure and CSP structure, which can effectively detect objects at multiple scales, and also does not have the adaptive anchor frame calculation and adaptive image scaling operation. For electrochemical image detection, the applicability to multi-scale objects is important to reduce the considerable computational effort for the next step of the denoising algorithm. The latter YOLOv7, despite its advantage in terms of the model parametric number, is too large and may overfit on simple object detection, i.e., it only remembers the locations where the images appear in the training set, instead of actually perceiving them, presenting a high Precision and lower Recall on this curve. YOLOv5 has an applicable network structure, which is computationally intensive compared to YOLOv7, which is smaller and performs better than YOLOv5 on multi-scale fusion.

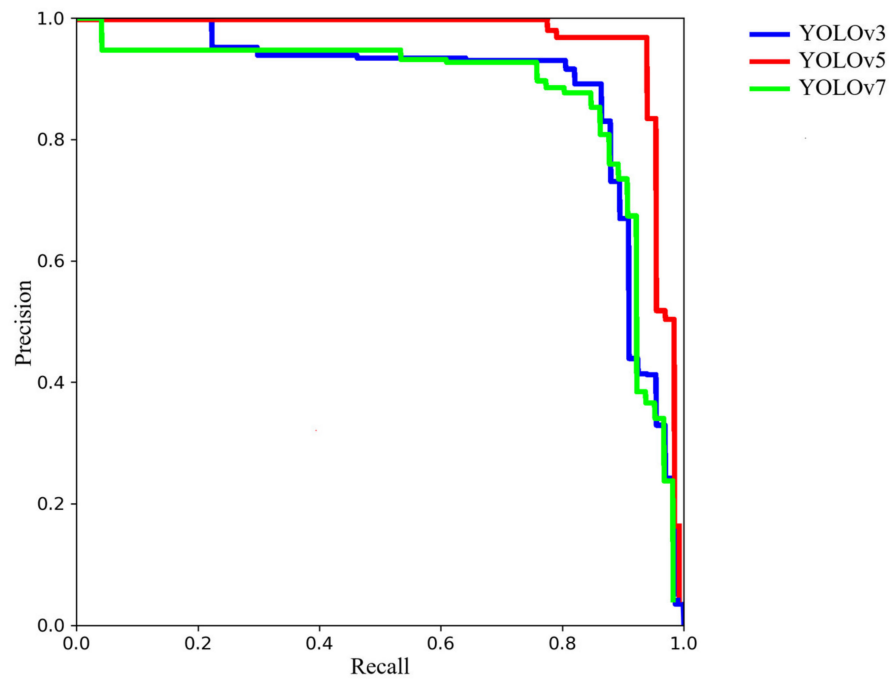


Figure 16. Precision-recall curve.

4.2. Comparison between CAMF and Other Filtering Algorithms

To evaluate the noise reduction effect and the light spot protection effect of CAMF and other commonly used filtering algorithms, we introduce three error metrics: Peak Signal-to-Noise Ratio (PSNR) [50], Image Enhancement Factor (IEF) [51] and Structural Similarity (SSIM) [50].

PSNR is an indicator to measure image quality, and its specific calculation steps are as follows:

Mean square error (MSE) is able to evaluate the discrepancy between a clean image and a restored image. MSE is defined as:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2 \tag{15}$$

where I represents a clean image and K represents a restored image, and both have size $m \times n$. Then, PSNR (dB) is defined as:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \tag{16}$$

Here, MAX_I^2 is the largest possible pixel value of the image. Because each pixel value is represented by an 8-bit binary in our experiment, MAX_I^2 is 255.

IEF is a measure of image noise reduction effect, which is defined as:

$$IEF(U, V, B) = \frac{\sum_{i=1}^m \sum_{j=1}^n (b_{ij} - u_{ij})^2}{\sum_{i=1}^m \sum_{j=1}^n (v_{ij} - u_{ij})^2} \tag{17}$$

where $U = [u_{ij}]$ is the clean image, $V = [v_{ij}]$ is the restored image, and $B = [b_{ij}]$ is the noisy image.

SSIM can measure the degree of distortion of images or the degree of similarity between two images from luminance, contrast, and structure. In this paper, we use clean

image x and restored image y to measure the degree of similarity. The specific formulas for the three degrees of SSIM are as follows:

$$l(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}, c(x, y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_1}, s(x, y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \quad (18)$$

Generally, $c_3 = c_2/2$. Here, μ_x represents the mean value of x , μ_y represents the mean value of y , σ_x^2 represents the variance of x , σ_y^2 represents the variance of y , σ_{xy} represents the covariance of x and y , $c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$ are two constants, and L is the biggest pixel value. Here, the value of L is 255, $k_1 = 0.01$ and $k_2 = 0.03$ as default values. So, SSIM is defined as:

$$SSIM(x, y) = [l(x, y)^\alpha \cdot c(x, y)^\beta \cdot s(x, y)^\gamma] \quad (19)$$

Setting α, β, γ to 1, we can obtain:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_1)} \quad (20)$$

For PSNR, IEF, and SSIM, the value of SSIM is between 0 and 1, while the value of PSNR and IEF can exceed 1. For a filtering algorithm, the higher the IEF value is, the better the noise reduction ability of the algorithm is, while the higher the corresponding value of PSNR or SSIM is, the stronger the algorithm is in protecting the light spot details of the image.

In the experiment, we implemented CAMF with python3.9, and we randomly selected 1000 images from the image dataset, which consists of ECL images made from a variety of different materials. In order to avoid the impact of the size of the images on the comparison results, we first cut these images into a uniform size of 3840×3840 , then the cropped images were processed by MF, IMF, MDBUTMF, NAFSMF, DAMF, AIFF, PDBF, and CAMF, respectively. Before using the above three error metrics to evaluate the results of each filtering algorithm, we asked Professor Lu Zhiwei from the College of Science at Sichuan Agricultural University in Sichuan Province to help us process the 1000 images, and the processed images were used as clean images for subsequent evaluation. In this paper, we refer to the images processed by each filtering algorithm as restored images and the unprocessed images we collected as noisy images. Then, we evaluated each filtering algorithm against these three metrics errors and figured out the corresponding PSNR, IEF, and SSIM values of each 1000 images after processing; the average values of PSNR, IEF, and SSIM of each filtering algorithm are shown in Table 1.

Table 1. Comparison of image processing effect between CAMF and the above filtering algorithms.

Filter Algorithms	PSNR (dB)	IEF	SSIM
IMF	30.59	341.74	0.935
MF	37.90	425.83	0.931
MDBUTMF	32.34	519.72	0.929
NAFSMF	31.98	434.61	0.922
DAMF	36.01	490.71	0.920
AIFF	34.23	407.77	0.919
PDBF	35.07	231.09	0.912
CAMF (Ours)	40.47	613.28	0.939

It can be seen from Table 1. that MF has a good processing effect, second only to CAMF, almost equal to CAMF in the score of SSIM, but behind CAMF in the score of IEF. From this, we can see that the light spot protection ability of MF is close to that of CAMF, but the denoising ability is far less than that of CAMF. It can be seen from the above table that the filtering algorithms improved based on MF for coping with SPN have lower scores

in PSNR and SSIM than MF, which can effectively prove the IMF, MDBUTMF, NAFSMF, DAMF, AIFF, and PDBF; six improved filtering algorithms based on MF are not suitable for images with irregular noise distributions such as ECL images. However, CAMF is ahead of other filtering algorithms in the three measures of PSNR, IEF, and SSIM. Additionally, CAMF far exceeds other filtering algorithms in the score of IEF, which proves that the denoising ability of CAMF is much higher than other filtering algorithms. In general, it can be considered that CAMF is superior to other algorithms in terms of retaining details and removing noise.

After evaluating each filtering algorithm against these three metrics errors, we also conducted experiments to compare the execution time of CAMF and other filtering algorithms. Similarly, we also randomly selected 1000 ECL images in the dataset, and then we processed them by using CAMF and other filtering algorithms respectively, and recorded the total time. Finally, we calculated the average values, as shown in Table 2.

Table 2. Comparison of execution time between CAMF and the above filtering algorithms.

Filter Algorithms	Time (s)
IMF	4.28
MF	3.70
MDBUTMF	5.69
NAFSMF	6.09
DAMF	1.23
AIFF	6.47
PDBF	4.50
CAMF (Ours)	4.02

As can be seen from Table 2, DAMF has the shortest execution time, followed by MF; CAMF has a slightly longer execution time than MF; IMF and PDBF have little difference in execution time; and MDBUTMF, NAFSMF, and AIFF all have execution times over 5 s.

Synthesizing Tables 1 and 2, we can see that although DAMF performs fast, it is far inferior to CAMF in noise reduction and light spot protection. We can also see that MF is faster than CAMF and has the same light spot protection as CAMF, but its noise reduction performance is far less than CAMF. We also see that the execution time of IMF and PDBF is similar. The light spot protection ability of IMF is not comparable to that of PDBF, but the noise reduction ability of IMF is better than that of PDBF. We find that the execution time of MDBUTMF, NAFSMF, and AIFF is more than 5 s, but the noise reduction ability of MDBUTMF is second only to CAMF, while the noise reduction ability and light spot protection ability of NAFSMF and AIFF are relatively poor, so they are not conducive to noise reduction in ECL images. Therefore, we can conclude that although the execution speed of CAMF is not the fastest, it is also relatively fast, and CAMF is far superior to other filtering algorithms in noise reduction and light spot protection.

A visual comparison of all the filtering algorithms is shown in Figure 17. As can be seen, CAMF is far better than other filtering algorithms in terms of its noise reduction ability and light spot protection ability, which can almost minimize the noise in the image without destroying the details of the image.

In order to ensure the generalization of CAMF, we further use two public datasets to continue the evaluation of all the above filtering algorithms. The first public dataset we use is a Fluorescence Microscopy Denoising (FMD) dataset "<https://www.cvmart.net/dataSets/detail/595> (accessed on 27 February 2023)". Some images from this dataset are shown in Figure 18. For the convenience of subsequent work, we crop all these images to 512×512 pixels.

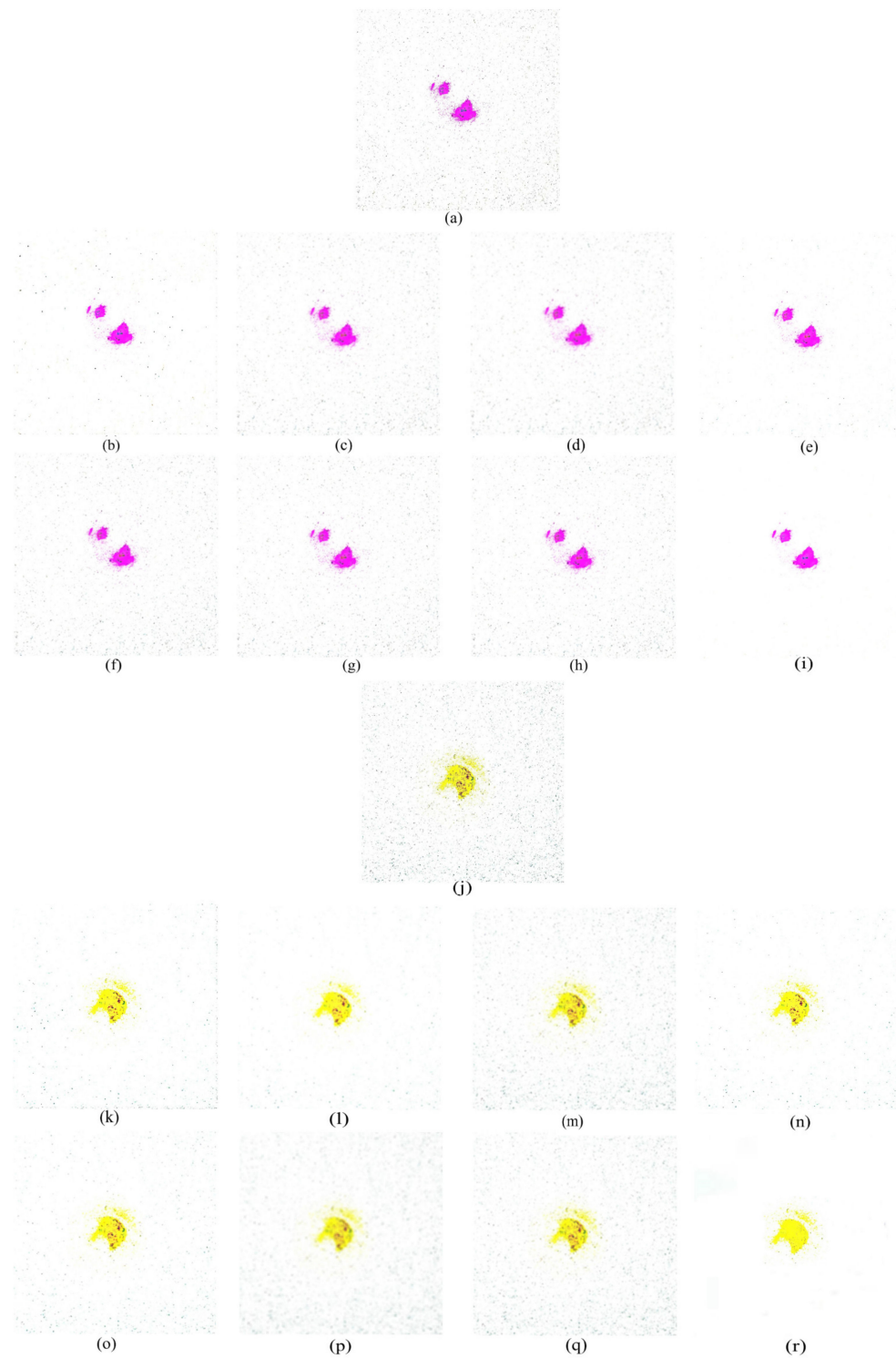


Figure 17. Visual comparison of all the filtering algorithms. (a) Noisy image 1; (b) DAMF; (c) PDBF; (d) AIF; (e) MF; (f) NAFSMF; (g) MDBUTMF; (h) IMF; (i) CAMF; (j) Noisy image 2; (k) DAMF; (l) PDBF; (m) AIF; (n) MF; (o) NAFSMF; (p) MDBUTMF; (q) IMF; (r) CAMF.

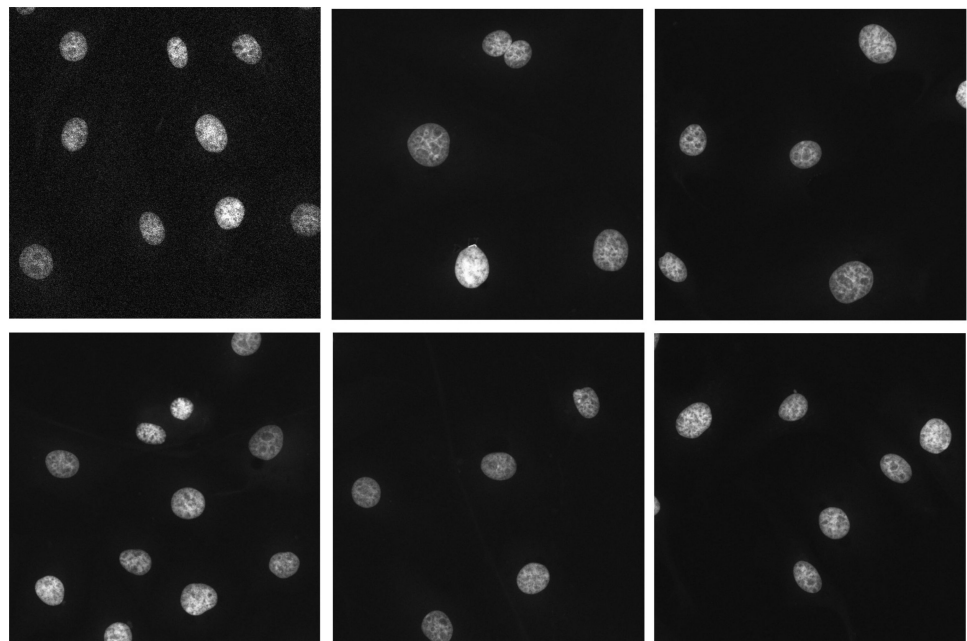


Figure 18. Some images from the FMD dataset.

The second public dataset we use is a Blood Cell (BC) image dataset “<https://www.cvmart.net/dataSets/detail/449> (accessed on 27 February 2023)”. Some images from this dataset are shown in Figure 19. Similarly, we crop all these images to 640×480 pixels.

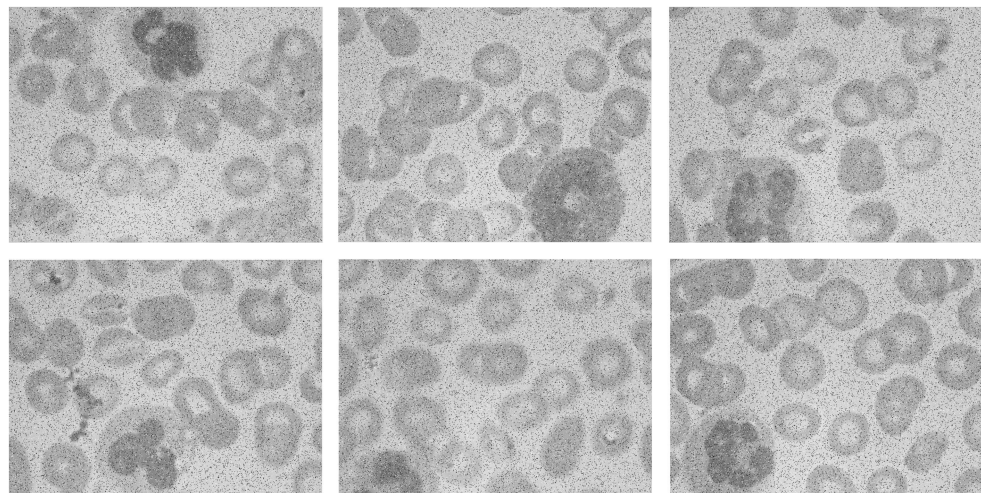


Figure 19. Some images from the blood cell image dataset.

We again evaluate CAMF based on the above two public datasets and three error metrics: PSNR, IEF, and SSIM. In this experiment, we selected 500 images from each of these two public datasets for the experiment. The specific compilation language, compilation environment, and experimental procedure are consistent with the above ECL image experiment. The eventual evaluation results of PSNR, IEF, and SSIM of all filtering algorithms are obtained as shown in Table 3.

Table 3. Evaluation results based on three error metrics: PSNR, IEF, and SSIM.

Dataset	Filter Algorithms	PSNR (dB)	IEF	SSIM
FMD	CAMF	40.61	589.73	0.932
Blood Cell	CAMF	39.68	609.73	0.933

We can see from Table 3. that the PSNR, IEF, and SSIM values of CAMF in FMD and Blood Cells are still high. The PSNR values are all above 39.00, the IEF values are all above 580.00, and the SSIM values are even above 0.930.

Synthesizing Tables 1–3, we can see that CAMF can not only be used for noise reduction in ECL images but also has a considerable effect on the processing of FMD and Blood Cell datasets. Moreover, it can be observed that the images in all three datasets have the same characteristics: fewer types of objects in the images and a simple background. Therefore, it is evident that CAMF is very good at processing images with these properties.

5. Conclusions

In order to achieve better performance of ECL image noise reduction, we propose CAMF for noise reduction. CAMF can not only change the size of the filter in real-time according to the different pixels to be processed but can also effectively adjust the size of the filter in real-time according to the position information between the current pixel and each light spot. It can solve the problem of MF, which cannot adjust the filter size in real-time and cannot simultaneously reduce noise and protect the details of the light spot. In the experiment, we used three datasets and three error metrics to evaluate the effect of CAMF and other filtering algorithms. In the three datasets, CAMF performs very well, with a PSNR greater than 39.00 dB and IEF greater than 580. It can be considered that CAMF can not only be used for noise reduction in ECL images but also has a certain generalization in other images.

Author Contributions: Conceptualization, J.Y., J.C. and J.L.; Methodology, J.Y.; Software, S.D. and Y.H.; Validation, J.Y., J.C. and Y.H.; Formal analysis, J.C.; Investigation, S.D. and Y.H.; Resources, J.Y.; Data curation, J.Y. and J.L.; Writing—original draft, J.Y. and J.C.; Writing—review and editing, J.Y., J.C. and J.L.; Visualization, S.D. and Y.H.; Project administration, J.Y.; Funding acquisition, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Sichuan Province Department of Education (Grant NO. JG2021-464).

Data Availability Statement: The data in this study are available on request from the corresponding author.

Acknowledgments: We are very grateful to Li Jun from the College of Information Engineering of Sichuan Agricultural University and Lu Zhiwei from the College of Science for their guidance. At the same time, we would like to thank Dai Shijie from the School of Science for providing the ECL dataset.

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

Acronyms

AIFF	Adaptive Iterative Fuzzy Filter
AMF	Adaptive Median Filter
AP	Average Precision
BC	Blood Cell
CAMF	Center-Adaptive Median Filter
CSP	Cross Stage Partia
CSPNet	Cross Stage Partial Network
DAMF	Different Applied Median Filter
ECL	Electrochemiluminescence
FMD	Fluorescence Microscopy Denoising
FPN	Feature Pyramid Network
IEF	Image Enhancement Factor
MDBUTMF	Modified Decision-Based Unsymmetric Trimmed Median Filter
MF	Median Filter
NAFSMF	Noise Adaptive Fuzzy Switching Median Filter
PAN	Path Aggregation Network

PDBF	Probabilistic Decision Based Filter
PSNR	Peak Signal-to-Noise Ratio
RepVGG	Reproducible Visual Geometry Group
SPN	Salt-and-Pepper Noise
SSIM	Structural Similarity
YOLO	You Only Look Once
YOLOv1	You Only Look Once version 1
YOLOv3	You Only Look Once version 3
YOLOv5	You Only Look Once version 5
YOLOv5s	You Only Look Once version 5 small
YOLOv7	vYou Only Look Once version 7
mAP	mean Average Precision

References

- Liu, S.; Yuan, H.; Bai, H.; Zhang, P.; Lv, F.; Liu, L.; Dai, Z.; Bao, J.; Wang, S. Electrochemiluminescence for Electric-Driven Antibacterial Therapeutics. *J. Am. Chem. Soc.* **2018**, *140*, 2284–2291. [[CrossRef](#)] [[PubMed](#)]
- Wang, Y.; Zhao, G.; Chi, H.; Yang, S.; Niu, Q.; Wu, D.; Cao, W.; Li, T.; Ma, H.; Wei, Q. Self-Luminescent Lanthanide Metal-Organic Frameworks as Signal Probes in Electrochemiluminescence Immunoassay. *J. Am. Chem. Soc.* **2021**, *13*, 504–512.
- Huang, W.; Hu, G.; Yao, L.; Yang, Y.; Liang, W.; Yuan, R.; Xiao, D. Matrix Coordination-Induced Electrochemiluminescence Enhancement of Tetraphenylethylene-Based Hafnium Metal–Organic Framework: An Electrochemiluminescence Chromophore for Ultrasensitive Electrochemiluminescence Sensor Construction. *Anal. Chem.* **2020**, *18*, 3380–3387. [[CrossRef](#)] [[PubMed](#)]
- Huo, X.-L.; Chen, Y.; Bao, N.; Shi, C.-G. Electrochemiluminescence integrated with paper chromatography for separation and detection of environmental hormones. *Sens. Actuator B-Chem.* **2021**, *334*, 129662. [[CrossRef](#)]
- Jafri, L.; Khan, A.; Siddiqui, A.; Mushtaq, S.; Iqbal, R.; Ghani, F.; Siddiqui, I. Comparison of high performance liquid chromatography, radio immunoassay and electrochemiluminescence immunoassay for quantification of serum 25 hydroxy vitamin D. *Clin. Biochem.* **2011**, *44*, 864–868. [[CrossRef](#)]
- Liu, Z.; Wang, L.; Liu, P.; Zhao, K.; Ye, S.; Liang, G. Rapid, ultrasensitive and non-enzyme electrochemiluminescence detection of hydrogen peroxide in food based on the ssDNA/g-C₃N₄ nanosheets hybrid. *Food Chem.* **2021**, *357*, 129753. [[CrossRef](#)] [[PubMed](#)]
- Peng, L.; Li, P.; Chen, J.; Deng, A.; Li, J. Recent progress in assembly strategies of nanomaterials-based ultrasensitive electrochemiluminescence biosensors for food safety and disease diagnosis. *Talanta* **2023**, *253*, 123906. [[CrossRef](#)]
- Hao, N.; Wang, K. Recent development of electrochemiluminescence sensors for food analysis. *Anal. Bioanal. Chem.* **2016**, *408*, 7035–7048. [[CrossRef](#)]
- Liu, Q.; Fei, A.; Wang, K. An immobilization-free and homogeneous electrochemiluminescence assay for detection of environmental pollutant graphene oxide in water. *J. Electroanal. Chem.* **2021**, *897*, 115583. [[CrossRef](#)]
- Han, Z.; Yang, Z.; Sun, H.; Xu, Y.; Ma, X.; Shan, D.; Chen, J.; Huo, S.; Zhang, Z.; Du, P.; et al. Electrochemiluminescence Platforms Based on Small Water-Insoluble Organic Molecules for Ultrasensitive Aqueous-Phase Detection. *Angew. Chem. Int. Ed. Engl.* **2019**, *58*, 5915–5919. [[CrossRef](#)]
- Busa, L.; Mohammadi, S.; Maeki, M.; Ishida, A.; Tani, H.; Tokeshi, M. Advances in Microfluidic Paper-Based Analytical Devices for Food and Water Analysis. *Micromachines* **2016**, *7*, 86. [[CrossRef](#)] [[PubMed](#)]
- Zhang, J.; Arbault, S.; Sojic, N.; Jiang, D. Electrochemiluminescence Imaging for Bioanalysis. *Rev. Anal. Chem.* **2019**, *12*, 275–295. [[CrossRef](#)] [[PubMed](#)]
- Diez-Buitrago, B.; Saa, L.; Briz, N.; Pavlov, V. Development of portable CdS QDs screen-printed carbon electrode platform for electrochemiluminescence measurements and bioanalytical applications. *Talanta* **2021**, *225*, 122029. [[CrossRef](#)]
- Zanut, A.; Fiorani, A.; Canola, S.; Saito, T.; Ziebart, N.; Rapino, S.; Rebecani, S.; Barbon, A.; Irie, T.; Josel, H.P.; et al. Insights into the mechanism of coreactant electrochemiluminescence facilitating enhanced bioanalytical performance. *Nat. Commun.* **2020**, *11*, 2668. [[CrossRef](#)] [[PubMed](#)]
- Brown, K.; Allan, P.; Francis, P.S.; Dennany, L. Psychoactive Substances and How to Find Them: Electrochemiluminescence as a Strategy for Identification and Differentiation of Drug Species. *J. Electrochem. Soc.* **2020**, *167*, 166502. [[CrossRef](#)]
- Zhang, Z.; Du, P.; Pu, G.; Wei, L.; Wu, Y.; Guo, J.; Lu, X. Utilization and prospects of electrochemiluminescence for characterization, sensing, imaging and devices. *Mater. Chem. Front.* **2019**, *3*, 2246–2257. [[CrossRef](#)]
- Chu, H.; Guo, W.; Di, J.; Wu, Y.; Tu, Y. Study on Sensitization from Reactive Oxygen Species for Electrochemiluminescence of Luminol in Neutral Medium. *Electroanalysis* **2009**, *21*, 1630–1635. [[CrossRef](#)]
- Zong, L.-P.; Ruan, L.-Y.; Li, J.; Marks, R.S.; Wang, J.-S.; Cosnier, S.; Zhang, X.-J.; Shan, D. Fe-MOGs-based enzyme mimetic and its mediated electrochemiluminescence for in situ detection of H₂O₂ released from Hela cells. *Biosens. Bioelectron.* **2021**, *184*, 113216. [[CrossRef](#)]
- Liu, T.; He, J.; Lu, Z.; Sun, M.; Wu, M.; Wang, X.; Jiang, Y.; Zou, P.; Rao, H.; Wang, Y. A visual electrochemiluminescence molecularly imprinted sensor with Ag+@UiO-66-NH₂ decorated CsPbBr₃ perovskite based on smartphone for point-of-care detection of nitrofurazone. *Chem. Eng. J.* **2022**, *429*, 132462. [[CrossRef](#)]

20. Zhang, Y.; Cui, Y.; Sun, M.; Wang, T.; Liu, T.; Dai, X.; Zou, P.; Zhao, Y.; Wang, X.; Wang, Y.; et al. Deep learning-assisted smartphone-based molecularly imprinted electrochemiluminescence detection sensing platform: Portable device and visual monitoring furosemide. *Biosens. Bioelectron.* **2022**, *209*, 114262. [[CrossRef](#)]
21. Goyal, P.; Khanna, N.; Dosad, J.; Gupta, M. Impact of neighborhood size on median filter based color filter array interpolation. *Math. Eng. Sci. Aerosp.* **2014**, *5*, 265–274.
22. Dong, H.; Zhao, L.; Shu, Y.; Neal, N. X-ray image denoising based on wavelet transform and median filter. *Appl. Math. Nonlinear Sci.* **2020**, *5*, 435–442. [[CrossRef](#)]
23. Ma, C.; Lv, X.; Ao, J. Difference based median filter for removal of random value impulse noise in images. *PLoS ONE* **2022**, *17*, e0264793. [[CrossRef](#)]
24. Wang, S.; Liu, Q.; Xia, Y.; Dong, P.; Luo, J.; Huang, Q.; Feng, D.D. Dictionary learning based impulse noise removal via L1–L1 minimization. *Signal Process.* **2013**, *93*, 2696–2708. [[CrossRef](#)]
25. Panetta, K.; Bao, L.; Agaian, S. A New Unified Impulse Noise Removal Algorithm Using a New Reference Sequence-to-Sequence Similarity Detector. *IEEE Access* **2018**, *6*, 37225–37236. [[CrossRef](#)]
26. Hwang, H.; Haddad, R.A. Adaptive median filters: New algorithms and results. *IEEE Trans. Image Process.* **1995**, *4*, 499–502. [[CrossRef](#)]
27. Wang, X.; Hou, R.; Gao, X.; Xin, B. Research on Yarn Diameter and Unevenness Based on an Adaptive Median Filter Denoising Algorithm. *Fibres Text. East. Eur.* **2020**, *28*, 36–41. [[CrossRef](#)]
28. Tripathy, S.; Swarnkar, T. Performance observation of mammograms using an improved dynamic window based adaptive median filter. *J. Discret. Math. Sci. Cryptogr.* **2020**, *23*, 167–175. [[CrossRef](#)]
29. Ahmed, F.; Das, S. Removal of High-Density Salt-and-Pepper Noise in Images With an Iterative Adaptive Fuzzy Filter Using Alpha-Trimmed Mean. *IEEE Trans. Fuzzy Syst.* **2014**, *22*, 1352–1358. [[CrossRef](#)]
30. Sheela, C.; Suganthi, G. An efficient denoising of impulse noise from MRI using adaptive switching modified decision based unsymmetric trimmed median filter. *Biomed. Signal Process. Control.* **2020**, *55*, 101657. [[CrossRef](#)]
31. Toh, K.K.V.; Isa, N.A.M. Noise Adaptive Fuzzy Switching Median Filter for Salt-and-Pepper Noise Reduction. *IEEE Signal Process. Lett.* **2010**, *17*, 281–284. [[CrossRef](#)]
32. Wang, Y.; Wang, J.; Song, X.; Han, L. An Efficient Adaptive Fuzzy Switching Weighted Mean Filter for Salt-and-Pepper Noise Removal. *IEEE Signal Process. Lett.* **2016**, *23*, 1582–1586. [[CrossRef](#)]
33. Lee, J.Y.; Jung, S.; Kim, P. Adaptive switching filter for impulse noise removal in digital content. *Soft Comput.* **2018**, *22*, 1445–1455. [[CrossRef](#)]
34. Erkan, U.; Gökrem, L.; Enginoğlu, S. Different applied median filter in salt and pepper noise. *Comput. Electr. Eng.* **2018**, *70*, 789–798. [[CrossRef](#)]
35. Deivalakshmi, S.; Palanisamy, P. Removal of high density salt and pepper noise through improved tolerance based selective arithmetic mean filtering with wavelet thresholding. *AEU-Int. J. Electron. Commun.* **2016**, *70*, 757–776. [[CrossRef](#)]
36. Balasubramanian, G.; Chilambuchelvan, A.; Vijayan, S.; Gowrisan, G. Probabilistic decision based filter to remove impulse noise using patch else trimmed median. *AEU-Int. J. Electron. Commun.* **2016**, *70*, 471–481. [[CrossRef](#)]
37. Sen, A.; Rout, N. Probabilistic Decision Based Improved Trimmed Median Filter to Remove High-Density Salt and Pepper Noise. *Pattern Recognit. Image Anal.* **2020**, *30*, 401–415. [[CrossRef](#)]
38. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
39. Mohiyuddin, A.; Basharat, A.; Ghani, U.; Peter, V.; Abbas, S.; Naeem, O.; Rizwan, M. Breast Tumor Detection and Classification in Mammogram Images Using Modified YOLOv5 Network. *Comput. Math. Methods Med.* **2022**, *2022*, 1359019. [[CrossRef](#)] [[PubMed](#)]
40. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
41. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-Style ConvNets Great Again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021; IEEE Trans: Nashville, TN, USA, 2021.
42. Wang, C.Y.; Bochkovskiy, A.; Liao, H. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
43. Khasawneh, N.; Fraiwan, M.; Fraiwan, L. Detection of K-complexes in EEG signals using deep transfer learning and YOLOv3. *Cluster Comput.* **2022**. [[CrossRef](#)]
44. Ali, L.; Alnajjar, F.; Parambil, M.; Younes, M.; Abdelhalim, Z.; Aljassmi, H. Development of YOLOv5-Based Real-Time Smart Monitoring System for Increasing Lab Safety Awareness in Educational Institutions. *Sensors* **2022**, *22*, 8820. [[CrossRef](#)] [[PubMed](#)]
45. Sozzi, M.; Cantalamessa, S.; Cogato, A.; Kayad, A.; Marinello, F. Automatic Bunch Detection in White Grape Varieties Using YOLOv3, YOLOv4, and YOLOv5 Deep Learning Algorithms. *Agronomy* **2022**, *12*, 319. [[CrossRef](#)]
46. Yang, Y. Drone-View Object Detection Based on the Improved YOLOv5. In Proceedings of the 2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, 25–27 February 2022; pp. 612–617.
47. Cao, M.; Fu, H.; Zhu, J.; Cai, C. Lightweight tea bud recognition network integrating GhostNet and YOLOv5. *Math. Biosci. Eng.* **2022**, *19*, 12897–12914. [[CrossRef](#)] [[PubMed](#)]
48. Chen, K.; Li, H.; Li, C.; Zhao, X.; Wu, S.; Duan, Y.; Wang, J. An Automatic Defect Detection System for Petrochemical Pipeline Based on Cycle-GAN and YOLO v5. *Sensors* **2022**, *22*, 7907. [[CrossRef](#)] [[PubMed](#)]

49. Wang, C.Y.; Liao HY, M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
50. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
51. Djurović, I. Combination of the adaptive Kuwahara and BM3D filters for filtering mixed Gaussian and impulsive noise. *Signal Image Video Process.* **2017**, *11*, 753–760. [[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.