

Reading Various Types of Pointer Meters Under Extreme Motion Blur

Hongyu Zhang^{ID}, Yunbo Rao^{ID}, Member, IEEE, Jie Shao^{ID}, Member, IEEE,
Fanman Meng^{ID}, Member, IEEE, and Jiansu Pu^{ID}

Abstract— Automatically reading pointer meters using deep learning has yielded promising results with high precision. However, existing methods ignore the interference with the camera brought by moving devices (e.g., patrol robots and drones), thus the persistent motion blur caused by the camera shake is not properly addressed. It is noteworthy that reading the pointer meter relies heavily on semantic segmentation of the scale and pointer within the meter. However, this can be challenging in extreme motion blur and diverse substation scenes. Moreover, reading various types of pointer meters and out-of-range pointer check remain tough issues. Thus, in this study, a full pipeline is proposed to solve the problems mentioned above. First, Filter-Deblur-U-net (FD-U-net) is proposed to ensure accurate segmentation under motion blur. To be specific, FD-U-net is a one-stage network consisting of a deblurring module and a segmentation module. The segmentation loss supervises the optimization of the deblurring module. And the proposed high-frequency residual attention (HFRA) in FD-U-net meticulously refines the details of motion-blurred image at the texture accumulated stage. Furthermore, the judgment-reading-algorithm (JRA) is developed to complete readings of 35 types of meters. To ensure practical application, we propose the data augmentation strategy called motion-blur-MixUp (MB-MixUp) to maintain precise meter localization under motion blur. Additionally, we propose a method called dark channel prior dehaze Laplace (DCPD-Laplace) to determine whether the meter patch is motion-blurred. Experimental results have demonstrated the whole pipeline achieves state-of-the-art performance with average relative error and average reference error of only 1.54% and 0.48%, respectively.

Index Terms— Image deblurring, motion blur, object detection, pointer meter recognition, semantic segmentation.

I. INTRODUCTION

VARIOUS types of pointer meters have been extensively used in numerous areas, such as substations and mechanical manufacturing. Meanwhile, numerous methods

Manuscript received 19 January 2023; revised 6 May 2023; accepted 10 June 2023. Date of publication 30 June 2023; date of current version 12 July 2023. This work was supported by the Science and Technology Project of Sichuan under Grant 2022ZHCG0033, Grant 2023ZHCG0005, and Grant 2023ZHCG0008. The Associate Editor coordinating the review process was Dr. Boby George. (*Corresponding author: Yunbo Rao.*)

Hongyu Zhang and Yunbo Rao are with the School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: Hong-yu-Zhang@outlook.com; raoyb@uestc.edu.cn).

Jie Shao and Jiansu Pu are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: shaojie@uestc.edu.cn; jiansu.pu@foxmail.com).

Fanman Meng is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: fmmeng@uestc.edu.cn).

Digital Object Identifier 10.1109/TIM.2023.3290962

incorporating computer vision algorithms in drones and patrol robots are becoming the mainstream in meter inspection. These methods are replacing manual meter reading, foot patrol inspection [1], as well as early fixed camera meter reading using computer vision [2]. Because the above approaches are less efficient, costly, and unsuitable for large substations inspections. In general, the existing algorithms deployed on the devices can be classified into two streams. One is the conventional computer vision method, and the other is the deep learning method. In brief, proper meter localization and feature extraction from the pointer and scale are essential for both streams of methods. However, the conventional computer vision methods are limited to handle meter with specific shape [3] and are still weak to localize the meters [4]. Deep learning methods typically use object detection networks, such as you only look once (Yolo) series and the region-based convolutional neural network (RCNN) series, to robustly locate the pointer meter. Subsequently, methods like conventional threshold segmentation of the pointer [5], semantic segmentation of the pointer and scale [6] and number coordinate acquisition using object detection [7] have been proved to be valid in meter reading. Moreover, combining localization and recognition in one network is a common approach in deep learning for simplifying the task pipeline. The one network pipeline methods include adopting key point detection and object detection simultaneously [8] or utilizing segmentation branches of the object detection network (i.e., mask R-CNN and its optimized versions) [9].

Since challenges and interference are common during outdoor patrol routes, some methods consider environmental factors such as rain, fog, and low illumination. Accordingly, data enhancement strategies like bilateral filter [10], and histogram equalization [11] can usually alleviate those outdoor interference. However, the above methods ignore the interference directly to the cameras of the device, especially when the real-time meter localization is required. In other words, the devices are usually in motion when capturing the image and sometimes generate extreme motion blur due to relative movement and camera shake, as presented in Fig. 1(a). In addition, the image quality of the meter is even worse when zooming and capturing the area of the meter, as shown in Fig. 1(b). There are many internal and external inducement of motion blur in inspection devices like drone and patrol robots. The cameras of drone devices can easily be affected by the airflow impact and the relative movement [12].

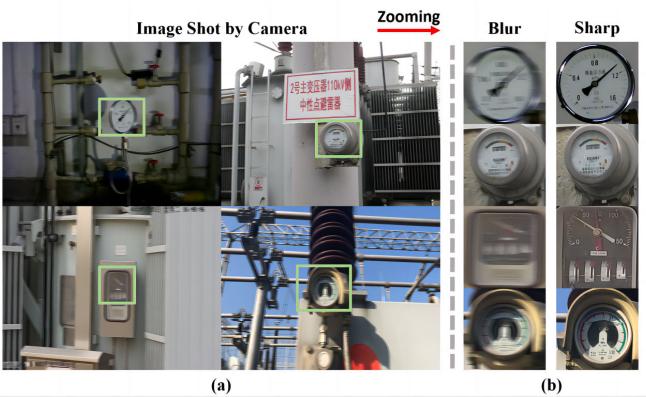


Fig. 1. Example images shot by devices in substation when motion blur occurs. (a) Image captured by devices with motion blur. (b) Contrast of motion-blurred meter image patches and sharp meter image patches after zooming to the area of the meter.

Fig. 2(a) presents the inspection drone under the unstable external condition. Moreover, patrol robots in motion might also fall into a similar dilemma when the ground is uneven, as illustrated in Fig. 2(b). The above situations usually generate motion-blurred images which severely influence semantic segmentation, key point detection, etc. Even with methods that can tolerate blurry images, decreasing reading accuracy is still unavoidable. In addition, upgrading the hardware of the existing devices to ensure more stable camera capturing process is unpractical due to the huge costs. Therefore, proposing an algorithm conducting pointer meter reading under the motion blur scene is necessary. Besides, to make the algorithm deployable, a motion blur judgment mechanism is also required to assert whether the captured image is motion-blurred. Moreover, most of the methods above ignore abnormal states of the meters (e.g., the pointer of the meter is out of the range), thus hindering the meter reading. Chao et al. [13] used a branch of their CNN model to judge whether a meter is out-of-range directly. However, this required more labeled data and is time-consuming when training a new classification branch. In summary, current methods still face multiple challenges. The complex background of the meters often hinders the localization of the meters. In addition, most existing methods still cannot read various types of meters. It is noteworthy that the camera of drones or patrol robots might be influenced by outdoor factors like airflow impact or uneven ground physically, thus obtaining a severe motion-blurred image, which hinders both localization and recognition of the pointer meters.

To more effectively address the above problems, a full pipeline under motion blur scene is designed. To improve meter detection accuracy under motion blur, a data augmentation strategy called MB-MixUp is proposed. And then dark channel prior dehaze Laplace (DCPD-Laplace) employing DCPD [14] is applied to judge whether the region of the meter is motion-blurred. Subsequently, Filter-Deblur-U-net (FD-U-net) and judgment-reading-algorithm (JRA) are developed to address various types of pointer meter readings in motion blur conditions while realizing out-of-range pointer check.

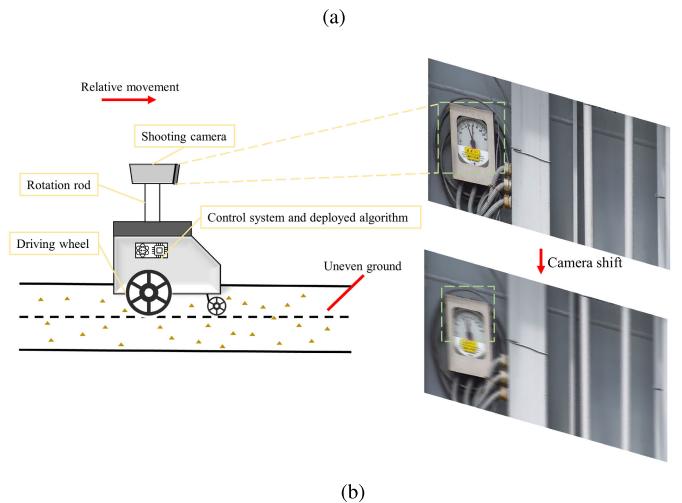
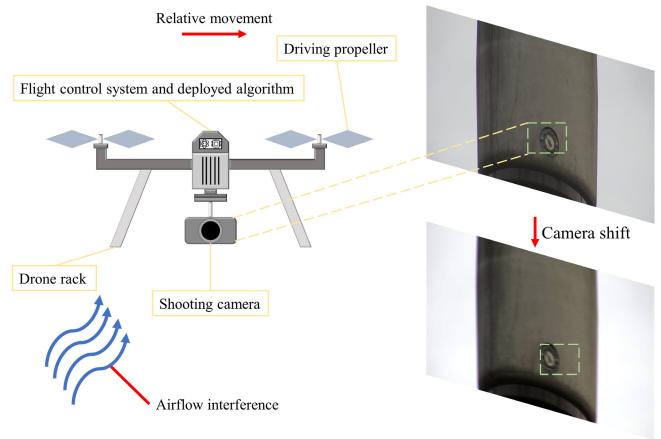


Fig. 2. Different devices conduct readings under complex and unstable environments with the risk of obtaining motion-blurred images. The red arrows imply unfavorable factors during meter inspection. (a) Drone encountering an airflow impact while reading meters. (b) Patrol inspection robot operating on the uneven ground.

The main contributions of this study can be stated from three perspectives.

- 1) A meter detection and motion blur judgment approach, leveraging the proposed MB-MixUp to assist the localization of the meter and determining whether the meter region is affected by motion blur using the proposed DCPD-Laplace.
- 2) A semantic segmentation model called FD-U-net can accurately segment the semantic features of meters, including pointers and scales, even when motion blur is present.
- 3) A robust meter reading method, named the JRA, is proposed. JRA can detect out-of-range pointers, and obtain reading results for various types of pointer meters.

The remainder of this study is organized into five sections. Section II overviews relevant previous studies. Section III describes the details of the proposed MB-MixUp, DCPD-Laplace, FD-U-net network, and JRA. Section IV demonstrates the implementation details of our experiments, and Section V illustrates the results of the proposed methods. Finally, Section VI summarizes this article and illustrates future expectations of this study.

II. RELATED WORK

In this section, the existing technique of automatic pointer meter reading using drones and patrol robots as well as motion deblurring networks is summarized. Furthermore, the problems to be solved are concluded.

A. Pointer Meter Reading in Machine Vision

The typical automatic reading algorithms in drone or patrol robot focus on positioning and feature extraction of the meter. In addition, the final pointer reading result is majorly based on the feature or information we have obtained. Conventional computer vision methods for reading meters require significant prior knowledge, such as the location of the scale's start and end [15]. Some methods are even designed specifically for a single type of meters [16]. Fortunately, more robust methods have been proposed for meter recognition. Gao et al. [17] cascaded a histogram of oriented gradients with a binary support vector machine (SVM) classifier to locate characters on the dashboard and conduct number recognition with multiclass SVM. Lv et al. [18] aligned the small-scale meters using a multilayer kernel regression positioning, and robust digital number recognition was conducted by using a multiclassifier under feature engineering. However, the above methods will encounter faults when the background of the meters is complex and cluttered or no template image is available. Zheng et al. [19] adopted multiscale retinex with color restoration to increase reading robustness in the complex environment but still can't deal with reading under motion blur.

Deep learning methods can achieve accurate and robust localization and recognition of meters, provided that the image is taken clearly. Pipelines of deep learning methods can be divided into two categories: pipelines with one network and pipelines with several networks.

For one network pipeline, Zuo et al. [20] used the mask R-CNN enhanced by precise roi pooling [21] to locate the meter and segment the dial and pointer of the meter simultaneously. Zheng et al. [8] and Howells et al. [22] have, respectively, proposed one network to locate the pointer and conduct key point detection. Chao et al. [13] proposed a multitask CNN to locate the meters and conduct readings.

For several networks pipeline, Li and Jia [6] employed Yolov3 to locate the meter, then segmented the information using the optimized U-net. Huang et al. [23] used an optimized Yolov3 and a monocular-vision pointer reconstruction algorithm based on speeded-up robust features to complete three types of pointer meters reading. In addition, Dong et al. [24] first utilized RCNN to locate the meter, then proposed a vector detection network to detect the tips of the pointer and the direction of the pointer at the same time using scalar maps. However, a template image was still needed to obtain the scale. Meanwhile, an improved Yolov4 with grayscale input proposed in [25] made localizing electrical equipment at night possible using the thermography camera. Fan et al. [26] used Yolov5 with a simplified global attention mechanism to locate the meter more precisely, and then segmented the pointer and obtained scale key points using a multitask U-net. Zhou et al. [7] applied Yolov5 at both

the localization and recognition stages to finish the pointer readings. In addition, A text detection method was proposed in [27] to locate the scale numbers on the meter and conduct readings using polar transform based on the text coordinates.

The methods above have achieved high accuracy of readings under their settings. However, they may have difficulty in processing different meters since some methods are significantly dependent on the specific feature of meters (i.e., text position and range of the scale). Most existing methods do not check for out-of-range pointers. Furthermore, current methods may struggle with feature extraction during the segmentation or key point detection stages, especially when there is severe motion blur due to camera shaking or relative movement of drones or patrol robots.

B. Motion Deblur and Motion Blur Judgment

As mentioned in previous studies, cameras are susceptible to recurring external interference. It is possible to generate motion-blurred images with very low quality, which will significantly hinder the process of feature extraction and meter reading. Moreover, improving the stability of patrol instruments in the environment with hardware can be expensive. Thus, it is essential to find a method in the algorithm to tackle the motion blur issue. Some methods have applied a deblurring algorithm in drones. Zhao et al. [12] used an optimized lucy-richardson algorithm with a Gaussian filter to deblur an image shot by drone. While Gong et al. [28] proposed a generative adversarial network (GAN) model using a scale-recurrent network as the generator in drone aerial image deblurring. Moreover, numerous recent studies have sought to restore an image significantly affected by motion blur. Multiinput and multioutput U-net (MIMO-U-Net) [29] rethought the coarse-to-fine strategy and designed an architecture with multiscale inputs of a blurred image and multiscale outputs of deblurred patches, using a u-shape structure with asymmetric feature fusion (AFF) to allow information flow from different scales. Comparably, Mao et al. [30] developed a plug-and-play deep residual Fourier transformation block (DeepRFT) based on MIMO-U-Net to restore information in the high-frequency domain. Other multistage methods like multistage progressive restoration (MPRNet) [31] progressively learn restoration function at the respective stage and pay more attention to the original resolution restoration. As vision transformer has been advancing, Wang et al. [32] proposed a general u-shape transformer using a locally-enhanced window transformer block to restore degraded images locally and globally. In addition, DeblurGANv2 [33] introduced feature pyramid networks [34] to the backbone of the GAN architecture. Subsequently, it employed clear images as the label to train a GAN model to deblur images. Moreover, some studies have applied motion deblur modules to other models. Zhou et al. [35] proposed a Filter-DeblurGAN to restore vehicle license plate images with motion blur. Zheng et al. [36] proposed a Deblur-Yolo which adds a pretrained Yolov3 to DeblurGAN to generate a clear detection-guided image for object detection on motion-blurred COCO datasets. However, for motion-blurred meter images, there is still a need for a network that can accurately segment

under motion blur. Furthermore, directly employing a deblurring network as preprocessing module complicates the overall pipeline and reduces efficiency.

Meanwhile, a mechanism to determine whether an image needs motion deblurring is required to increase efficiency and deploy in practice. Most existing methods are no-reference, meaning that a custom threshold is required. Zhou et al. [35] applied the Laplace operator on the input image to get the edge information value of the image. If the value is below a certain threshold, they can assert the image is blurred. Said et al. [37] summarized the focus measurement operator and a judgment method using the variance of the Laplace operator is introduced. Other methods like diagonal Laplace proposed in [38], and gradient-based tenengrad function mentioned in [39] have also demonstrated some degree of performance in no-reference blur judgment. However, a limitation of the methods above is that they cannot distinguish between different types of blurriness. Specifically, effects that will not cause feature extraction failures like slight mist, out-of-focus background, and blur effect caused by imbalance illumination [40] can confuse the judgment mechanism and take the clear images as blur images. Therefore, a more accurate judgment mechanism design for motion blur is needed.

III. METHODOLOGY

In this section, the proposed MB-MixUp, DCPD-Laplace, FD-U-net, and JRA are elucidated to address the above problems more effectively. The data augmentation strategy MB-MixUp is introduced first. Subsequently, we provide a detailed explanation of the detailed principle of DCPD-Laplace. And then we introduce the network structure and loss function of FD-U-net. Finally, abnormal pointer check and multitype meter reading methods of the JRA are discussed. Fig. 3 presents the workflow of the overall pipeline.

A. Motion Blur MixUp

Precise localization of the pointer meters is the prerequisite to ensure accurate pointer meter readings. Unfortunately, motion blur might interfere with the meter detection process, especially when identifying meter types. Consequently, a data augmentation strategy called MB-MixUp is developed to assist the detector (e.g., Yolov5s). In our case, pixel changes caused by motion blur in the object area often impede accurate recognition of the pointer meter, such that more misclassification issues occur, and the confidence score may be reduced. In addition, meter in a far local length may directly result in miss detection. Since adding subnetwork which can address motion blur in the case of high-level task like object detection is costly, this study attempts to find a concise way to facilitate meter detection under motion blur at the data enhancement stage, while no extra inference time is added.

Specifically, MB-MixUp is designed to roughly simulate the motion blur based on human vision, as presented in Fig 4. The motion blur simulation can be conducted by simulating two processes: relative movement and out-of-focus.

To simulate the relative movement of the camera, the pointer meter image patch I is first located with the annotation files.

And then I is translated with a random angle to obtain m translated images, as expressed in the following equation:

$$I_{\text{translate}}^i = \begin{bmatrix} 1 & 0 & S_x^i \\ 0 & 1 & S_y^i \end{bmatrix} \times I \quad (1)$$

where $I_{\text{translate}}^i$, $i \in \{1, 2, 3, \dots, m\}$ is the image after translation. S_x^i and S_y^i are the offset along x - and y -axis, respectively, and they are generated randomly from the interval, as illustrated in the following equation:

$$S_x^i, S_y^i \in \left(-\frac{1}{s} \min(H, W), \frac{1}{s} \min(H, W) \right) \quad (2)$$

where H and W denote the height and width of I , respectively. And s represents the scale factor, which is set as two in our study. Next, we perform a weighted addition of I and $I_{\text{translate}}^i$. Finally, the mean filter is applied to simulate the out-of-focus phenomenon, as presented in the following equation:

$$I_{\text{enhanced}} = K * \left(\sum_{i=1}^m \alpha_i \cdot I_{\text{translate}}^i + \left(1 - \sum_{i=1}^m \alpha_i \right) \cdot I \right) \quad (3)$$

where I_{enhanced} is the final enhanced image, and K denotes the mean filter. α_i is the weight factor for $I_{\text{translate}}^i$. In this study, α_i is uniformly set to 0.1, and m is set to two. At last, the final step of MB-MixUp involves replacing the area occupied by I with I_{enhanced} according to the annotation files.

B. DCPD-Laplace Motion Blur Judgment Mechanism

We aim to develop a precise judgment mechanism for motion-blurred images. It requires the ability to distinguish other noises that reduce image sharpness but do not affect semantic segmentation, such as mist or out-of-focus backgrounds. Thus, the DCPD-Laplace is proposed using the Laplace operator with DCPD to filter images that should be deblurred.

The fog degradation model of DCPD is first introduced, as expressed in the following equation:

$$I(x) = t(x) \cdot J(x) + (1 - t(x)) \cdot A \quad (4)$$

where $J(x)$ denotes the noise free image, $t(x)$ is the transmission map of the image, A is the global atmospheric light, and $I(x)$ is the observed image.

DCPD is sufficient in dealing with mist-like or smooth-filter-like noise that is often confused with motion blur. It aims at obtaining the transmission map $t(x)$ and global atmospheric light A of the image. It proves a prior knowledge of haze-free outdoor images: At least a color channel has very low intensity at some pixels in most of the nonsky patches. This can be formulated in the following equation [14]:

$$J^{\text{dark}}(x) = \min_{c \in \{r, g, b\}} \left(\min_{y \in \omega(x)} J^c(y) \right) \approx 0 \quad (5)$$

where J^c expresses one of the channels in J , $\omega(x)$ is the local patch centered at x , and J_{dark} is the pixel intensity in the dark channel which approximates 0. The top 0.1% brightest pixels are obtained in the dark channel map of $I(x)$, and A is gained by calculating the average of those pixels in $I(x)$. After A is calculated, $I(x)$ is transformed into its' dark channel using

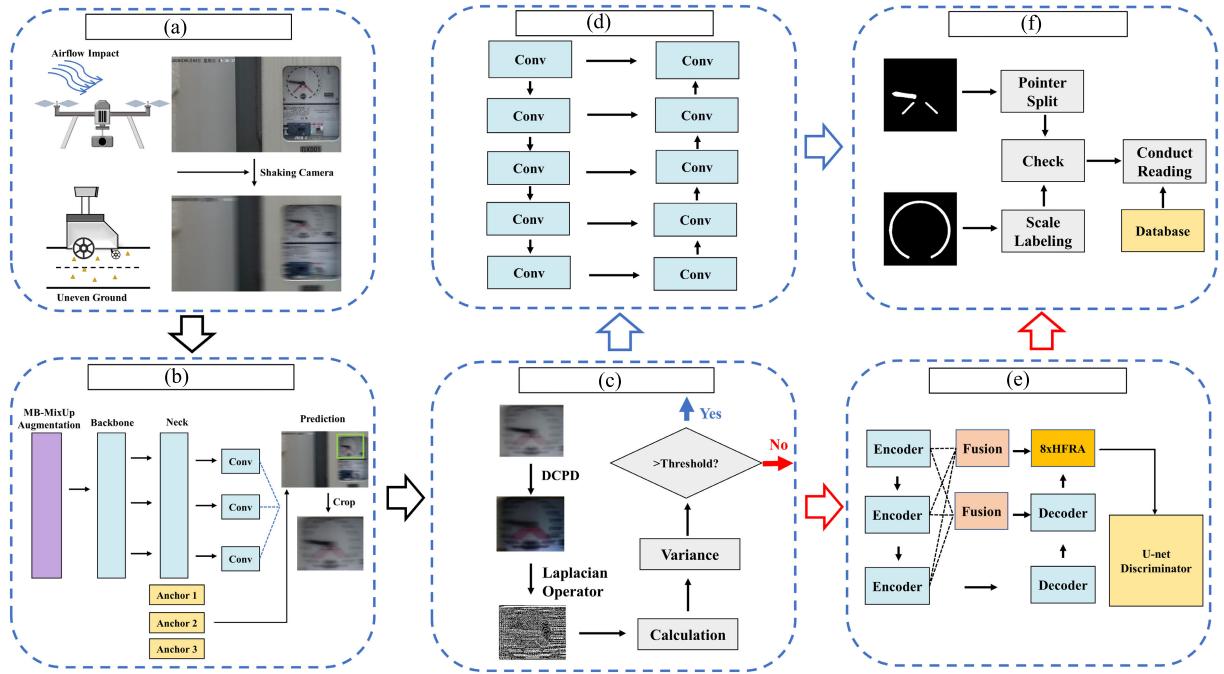


Fig. 3. Overall process of the proposed pipeline. (a) Devices taking images under unstable and complex environments. (b) Utilize Yolov5s enhanced by MB-MixUp to obtain the meter location. (c) Judge whether the image patch is motion-blurred using DCPD-Laplace. (d) Conduct the segmentation of pointer and scale using FD-U-net. (e) Conduct the segmentation of pointer and scale using FD-U-net. (f) Start meter reading using JRA. The initial steps of the pipeline are (a)→(b)→(c). If the image is not motion-blurred judged by DCPD-Laplace, the following steps are (c)→(d)→(f) as the blue arrow. If the image is motion-blurred, the following steps are (c)→(e)→(f) as the red arrow.

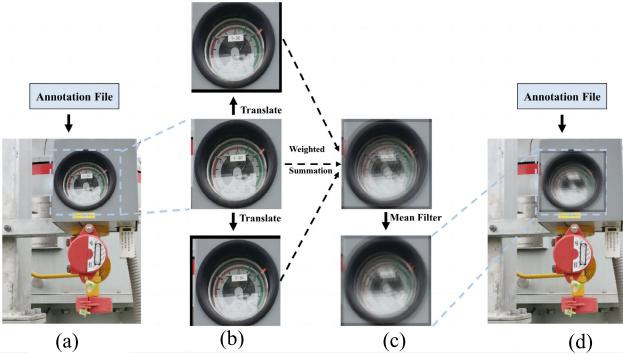


Fig. 4. Illustration of MB-MixUp. (a) Original image at training phase. (b) Crop out the object patches and then obtain translated patches. (c) Image patches fusion and blur. (d) Enhanced image.

(5). Then the transmission map $t(x)$ is calculated with A as in the following equation [14]:

$$t(x) = 1 - \min_{c \in \{r, g, b\}} \left(\min_{y \in \Omega(x)} \frac{I^c(y)}{A^c} \right). \quad (6)$$

Eventually, the dehazed image $J(x)$ is approximately calculated according to $t(x)$ and A .

The DCPD is capable of effectively removing the nonmotion-blur factors that reduce the value of edge detection or gradient operator. On that basis, the accurate motion blur degree of the image can be determined using the Laplace convolution kernel, as shown in the following equation:

$$\Delta = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}. \quad (7)$$

Finally, we summarize the process of DCPD-Laplace as formulated in the following equation:

$$T = \sum_{(x,y) \in I} \left(\Delta * I^{\text{dehaze}}(x, y) - \overline{\Delta * I^{\text{dehaze}}} \right)^2 \quad (8)$$

where $I^{\text{dehaze}}(x, y)$ denotes the pixel intensity at (x, y) after DCPD, Δ denotes the Laplace operator, $\Delta * I^{\text{dehaze}}$ is the mean values of Laplacian image of I^{dehaze} . If T is smaller than the predefined threshold, the image will be sent to FD-U-net, and if not, it will directly go to a conventional segmentation network such as U-net.

C. Network Structure of FD-U-Net

After the implementation of the DCPD-Laplace motion blur judgment mechanism, the actual motion-blurred image is input into our designed network FD-U-net. The proposed FD-U-net is a one-stage network conducting segmentation in the extreme motion blur condition. In the field of automatic meter reading, the computation complexity and time-sensitive problems cannot be ignored. Most CNN-based image deblurring methods require numerous subnetworks stacked to each other. Although they achieve good performance in image deblurring, they are time-consuming. Inspired by the structure of MIMO-U-Net, we design a three-scale motion deblurring module and stack a U-net discriminator to supervise the deblurring effect according to the pointer and scale segmentation loss. The reading according to the segmentation result of our FD-U-net will prove to be more precise compared to other methods with two networks by first deblurring the image and then utilizing

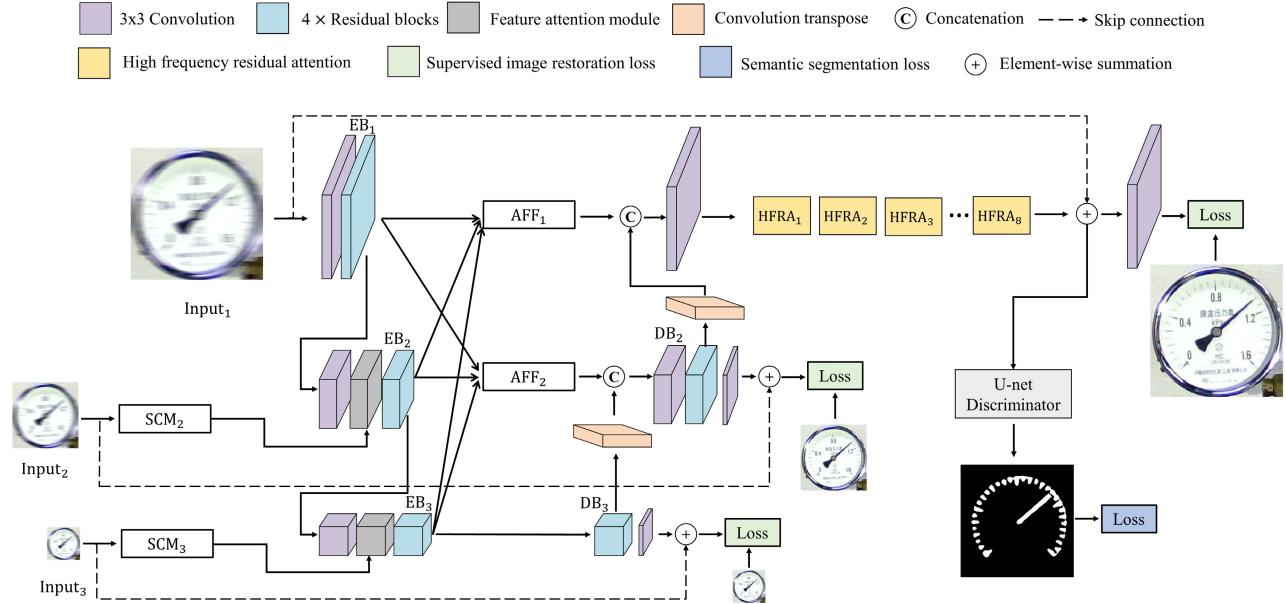


Fig. 5. Overall architecture of the FD-U-net. It consists of three parts. First, multiinput single encoder and multioutput single decoder. Second, detailed texture refine with HFRA at original resolution. Third, the U-net discriminator head.

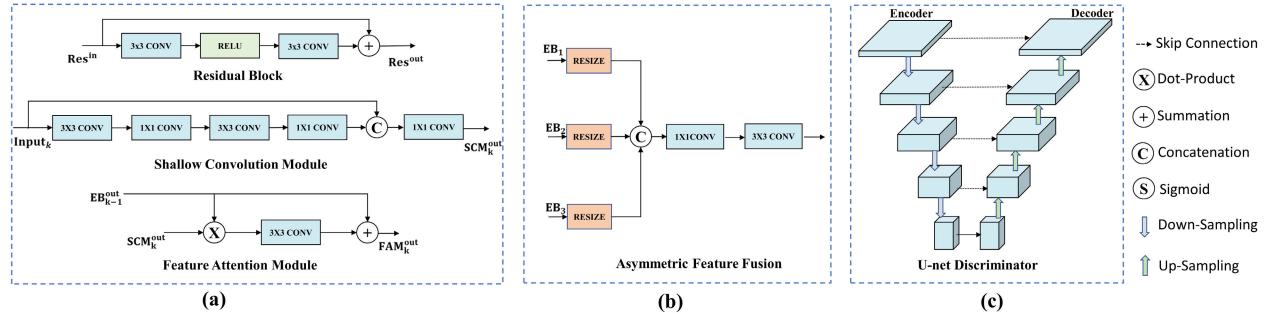


Fig. 6. Detailed basic network architecture of FD-U-net. (a) Basic modules of EB and DB: residual block, SCM, and FAM. (b) Details of AFF module. (c) Brief structure of U-net discriminator.

segmentation. The overall architecture of FD-U-net is shown in Fig. 5. The detailed network is exploited as follows.

1) *Multiinput Single Encoder and Multioutput Single Decoder*: We employ light-weight multiinput encoder and multioutput decoder to roughly refine the blurred image. In the encoder blocks (EBs), we let H and W denote the height and width of the input image, respectively. The input image is rescaled into three stages [i.e., (H, W) , $((1/2)H, (1/2)W)$, and $((1/4)H, (1/4)W)$], which are expressed as input_k , $k \in \{1, 2, 3\}$. Besides, encoders for each of them are represented as EB_1 , EB_2 , EB_3 , and outputs of the EBs are denoted as EB_k^{out} , $k \in \{1, 2, 3\}$. For input_1 , EB_1^{out} is straightforward to be obtained by passing through a 3×3 convolution and four residual blocks. While input_2 and input_3 are first refined by the shallow convolution module (SCM) and then passed through the feature attention module (FAM) immediately followed by four residual blocks. As shown in Fig. 6(a), the SCM applies two 1×1 stacked with 3×3 convolutions on the input image. Next, the original input is concatenated and then passed through one 1×1 convolution to fuse the shallow feature. The k th level output of SCM is denoted as $\text{SCM}_k^{\text{out}}$. As illustrated in Fig. 6(a), $\text{SCM}_k^{\text{out}}$ is then

fed into FAM, which performs a dot product with the output of the previous stage EB (i.e., $\text{EB}_{k-1}^{\text{out}}$). The resulting tensor is then passed through a 3×3 convolution and added elementwise to $\text{EB}_{k-1}^{\text{out}}$ in order to learn complementary features between the upper and lower stages, which can be formulated as $\text{FAM}_k^{\text{out}}$. Finally, four residual blocks are set further to refine the feature of $\text{FAM}_k^{\text{out}}$, thus obtaining EB_k^{out} , $k \in \{2, 3\}$. As shown in Fig. 6(b), the AFF module is subsequently employed to first resize $\{\text{EB}_1^{\text{out}} - \text{EB}_3^{\text{out}}\}$ to the specific scale for different stages [i.e., (H, W) , $((1/2)H, (1/2)W)$]. Subsequently, they are concatenated together and passed through a 1×1 convolution and a 3×3 convolution. The results are denoted as AFF_1 and AFF_2 with the scale of (H, W) and $((1/2)H, (1/2)W)$, respectively.

In the decoder blocks (DBs): the input of DB_3 is directly EB_3^{out} . Additionally, the input to DB_2 is obtained by concatenating the output of AFF_2 with the result of upsampling DB_3 using transpose convolution. By applying one 3×3 convolution, four residual blocks, and another 3×3 convolution to the inputs of $\{\text{DB}_2 - \text{DB}_3\}$, we obtain the output feature map of DB_2 and DB_3 . The output of DB_2 and DB_3 , with the addition of input image at each stage, are supervised

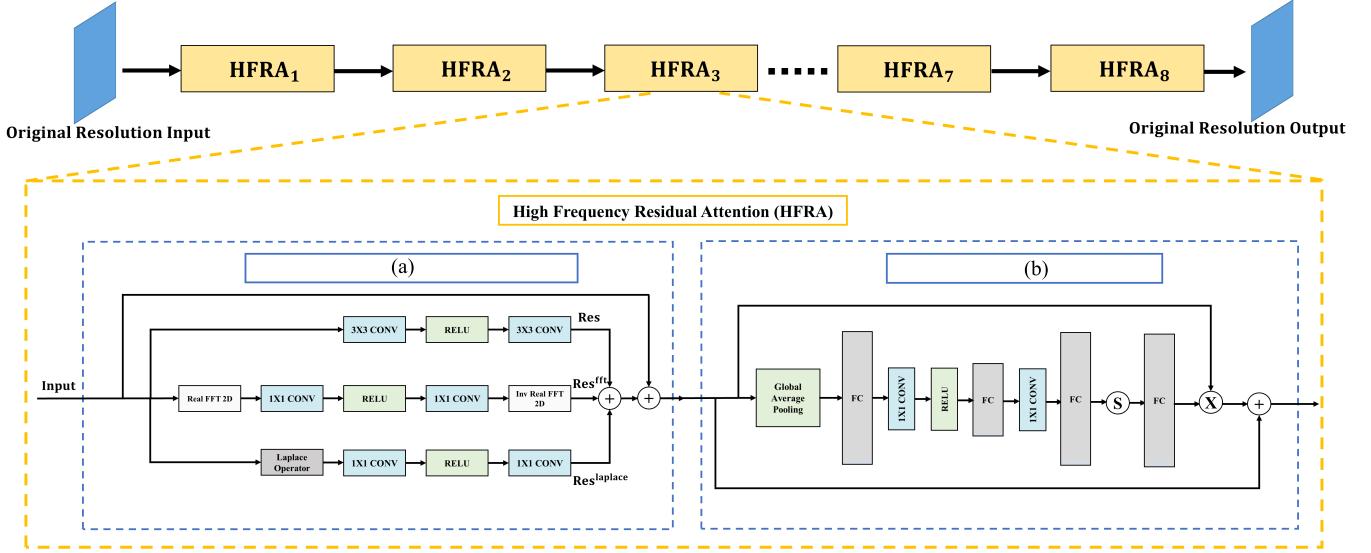


Fig. 7. Original resolution stage refined and detailed architecture of our proposed HFRA. (a) Structure of HFRB in HFRA. (b) Structure of CAB in HFRA.

by the sharp images with the corresponding scale (i.e., GT₂, GT₃, respectively). Subsequently, we concatenate the output of AFF₁ with the result of upsampling DB₂ using transpose convolution to obtain the input for the high-frequency residual attention (HFRA), which will be discussed in Section III-C2.

2) High-Frequency Residual Attention: The original MIMO-U-net's DBs at different stages are endowed with equal importance. However, when the feature map is restored to the original resolution of GT₁, there are more texture details in the feature map. Therefore, simply using eight residual blocks to refine the above feature maps may result in inadequate learning of the high-frequency information, thus careful refine at the original stage is required. In our case, the edge information of a pointer meter is enormous between its black scale and the white dial, which requires more detailed attention and refinement. Thus, HFRA blocks are proposed to further restore the detailed information at the original resolution after the rough refine of encoders and decoders.

As depicted in Fig. 7, the HFRA consists of high-frequency residual block (HFRB) and channel attention block (CAB) [41]. Inspired by the strategy to refine the image from the frequency domain in [30], the HFRB utilizes a fast Fourier transform (FFT) patch and Laplace patch to assist the restoration of the edge information in the residual block from both the spatial and frequency domains. As presented in Fig. 7, we denote the input feature map as Input. And then the output of the original residual patch is formulated as Res. Subsequently, the FFT patch will first transform the spatial feature map into the frequency domain using 2-D FFT. And then two 1 × 1 convolutions are applied to further learn the information in the frequency domain. Finally, 2-D inverse FFT is adopted to transform the feature map back to the spatial domain, and the output is expressed as Res^{fft}. Similarly, the Laplace patch first employs a Laplace operator to the feature map and then passes through two 1 × 1 convolutions to obtain the output of the patch, denoted as Res^{laplace}. Subsequently, we elementwisely add Input,

Res, Res^{fft} and Res^{laplace} to obtain the output of the HFRB and then pass it to the CAB, which further strengthens the feature map by selectively highlighting critical feature in the channel dimension using an attention map. By stacking eight HFRA blocks together, our method enables fine-grained restoration of textual details in images, which is crucial for optimizing the performance of the FD-U-net.

3) U-Net Discriminator: The U-net discriminator is primarily used to obtain the final segmentation masks. In addition, the segmentation loss of the U-net discriminator provides supervision for the deblurring module. This encourages the deblurring module to generate feature maps that are more suitable for segmentation. However, using a U-net [42] without training can lead to model oscillation, making it difficult to achieve convergence. Therefore, the U-net discriminator is pretrained on half of our clear meter segmentation dataset. During the training phase, the segmentation loss will backpropagate to the HFRA blocks, multistage encoders, and decoders to update the parameters. The brief architecture of U-net discriminator can refer to Fig. 6(c).

4) Loss Function: I_k denotes the output of k th deblurring stage in FD-U-net. Correspondingly, G_k denotes the k th ground truth clear image. M represents the number of the deblurring stage. The parameters of FD-U-net are updated by the loss function as follow.

The multiscale content L1 loss function [33] is set to directly supervise the differences in pixels between deblurred and ground truth image, as expressed in the following equation:

$$L_{\text{content}} = \sum_{k=1}^M \frac{1}{n_k} \|I_k - G_k\|_1 \quad (9)$$

where n_k denotes the quantity of pixels in the k th stage image, and $\|\cdot\|$ denotes the L1 loss function.

Moreover, the multiscale frequency reconstruction (MSFR) in [29] is adopted to supervise the high-frequency information restoration in the frequency domain, as illustrated in the

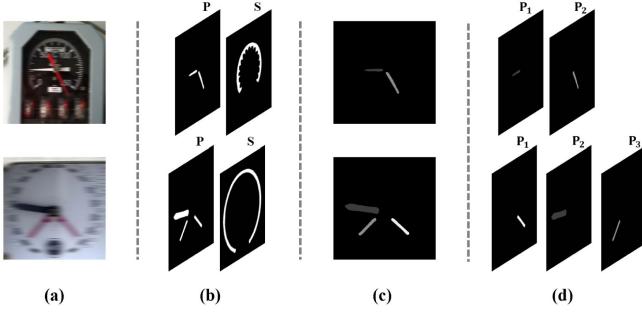


Fig. 8. Segmentation masks of FD-U-net and pointers split. (a) Motion-blurred image. (b) Segmentation masks of FD-U-net. (c) Result of connected component labeling. (d) Result of multiple pointer splitting.

following equation:

$$L_{MSFR} = \sum_{k=1}^M \frac{1}{n_k} \|F(I_k) - F(G_k)\|_1 \quad (10)$$

where F refers to the process of FFT.

At the same time, to reduce edge loss in the spatial domain, the Laplacian operator is utilized to convey the edge loss, as presented in the following equation:

$$L_{edge} = \sum_{k=1}^M \frac{1}{n_k} \|\Delta(I_k) - \Delta(G_k)\|_1 \quad (11)$$

where Δ denotes the Laplacian operator. Finally, the Focal loss [43] is applied to supervise the segmentation of FD-U-net, as expressed in the following equation:

$$L_{segmentation} = \text{Focal}(\text{Mask}, \text{Mask}_{GT}) \quad (12)$$

where Focal denotes the Focal loss, Mask represents the output segmentation mask, and Mask_{GT} denotes the ground truth segmentation mask.

After balancing the four types of loss functions, the overall loss function of the FD-U-net is summarized in the following equation:

$$L = L_{content} + \lambda_1 L_{MSFR} + \lambda_2 L_{edge} + L_{segmentation} \quad (13)$$

where λ_1 is set to 0.2, and λ_2 is set to 2 in consideration of a large amount of edge information in the clear pointer meter image. The special attention to the details information and segmentation supervision on the deblurring module will contribute to the global optimization of FD-U-net.

D. Judgment Reading Algorithm

JRA leverages the segmentation mask generated by FD-U-net to read the pointer meter. Specifically, JRA first divides the pointer mask into multiple individual pointer masks. It then assigns labels to each scale using the virtual rotation pointer (VRP) and performs an out-of-range check before obtaining the final pointer readings.

1) Multi Pointer Split and Mathematization of the Masks: Since a pointer meter might have over one pointer, and the segmentation masks of a pointer meter are only two: pointer mask (P) and scale mask (S) as presented in Fig. 8(b). The first step in the process is to distinguish between multiple

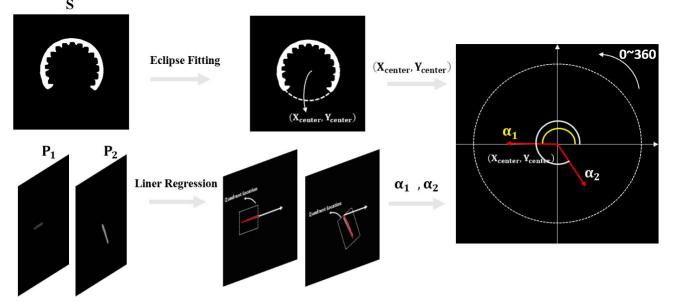


Fig. 9. Mathematization of key information in the segmented masks. Eclipse fitting is applied to obtain the center of the meter: (X_{center}, Y_{center}) . Then utilize linear regression to obtain the rotation angles of the pointers.

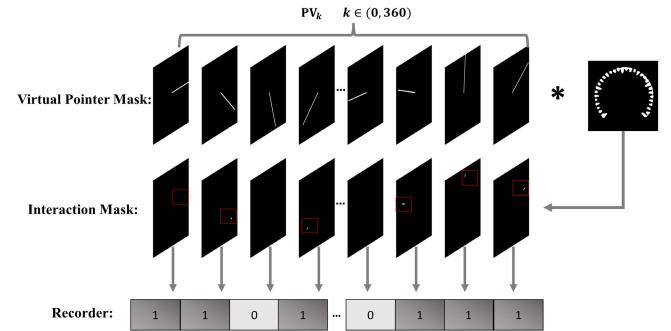


Fig. 10. Scale labeling process using VRP. The first row is the process of PV_k where $k \in (0, 360)$ dot-producting with S . The second row illustrates the result of the dot-product. The third row is the final result of Recorder.

pointers in P . To achieve this, we first erode P to reduce the noise pixels. We then employ connected component labeling to identify multiple pointers in P , as illustrated in Fig. 8(c). In the next step, each individual pointer is split from P . This process is illustrated in Fig. 8(d), where P_i represents a single pointer extracted from P . To further get the key information for readings, the least square eclipse fitting is adopted on S to obtain the approximate center of the pointer meter: (X_{center}, Y_{center}) . Then for each $P_i \in P$, a linear regression is employed to obtain the slope of the single pointer. To determine the rotation angle of each pointer, we use a counterclockwise convention from the positive direction of the x -axis, with the rotation angle increasing from 0° to 360° . To obtain the quadrant that each pointer belongs to, we calculate the distribution of P_i at our coordinates. With this information, we can determine the rotation angle of P_i , denoted as $\alpha_i \in \alpha$, based on the quadrant and the slope, as illustrated in Fig. 9.

2) Scale Labeling Using VRP: After obtaining the α , the crucial issue is asserting which part of the fitting ellipse has the scale. The proposed VRP is adopted to label whether a part of the fitting ellipse has the scale. Specifically, a new mask denoted as PV_k is created, where $k \in (0, 360)$ represents the rotation angle of the virtual pointer will be placed on PV_k . And then the virtual pointer can be expressed in the following equation:

$$\begin{cases} (x, y)_{start} = (X_{center}, Y_{center}) \\ (x, y)_{end} = (x, y)_{start} + \left(\frac{\sqrt{2}W}{2} \cos(k), \frac{\sqrt{2}H}{2} \sin(k) \right) \end{cases} \quad (14)$$

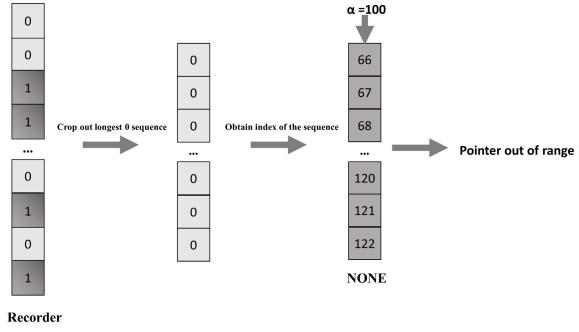


Fig. 11. Brief flowchart of the pointer out-of-range check.

where $(x, y)_{\text{start}}$ and $(x, y)_{\text{end}}$ represent the start and end point of the virtual pointer, respectively, and W, H denote the width and height of the image. To reduce errors, each virtual pointer occupies only one pixel. Then a dot-product is conducted between every PV_k and S . We use a list named Recorder to record the distribution of scales at each rotation angle k . Specifically, $\text{Recorder}[k] = 1$ indicates the presence of a scale at rotation angle k , while $\text{Recorder}[k] = 0$ indicates the absence of a labeled scale. And the judgment method can be referred in the following equation:

$$\begin{cases} \text{Recorder}[k] = 1, & PV_k \cdot S > 0 \\ \text{Recorder}[k] = 0, & PV_k \cdot S = 0, \quad k \in (0, 360). \end{cases} \quad (15)$$

If $PV_k \cdot S > 0$, it indicates that there is an interaction between these two masks, and therefore there is a scale at rotation angle k . Conversely, if $PV_k \cdot S \leq 0$, it means that there is no interaction between these two masks, and therefore there is no scale at rotation angle k , as presented in Fig. 10.

3) *Out-of-Range Check and Multi Pointer Reading*: After obtaining Recorder, we can check whether a pointer is out-of-range. Since there might exist some small intervals in the scale mask, the longest continuous sequence where $\text{Recorder}[k]$ is equal to 0 should be found, which indicates the real no scale area in the actual meter. Then the subscript set of this continuous sequence is extracted from $\text{Recorder}[k]$ and is expressed as NONE. If $\exists \alpha_i \in \alpha$ in NONE, it is suggested that the pointer is out-of-range, as illustrated in Fig. 11. After NONE is generated, the reading can be conducted by first acquiring the angle of the start scale denoted as s_{\min} using the last element of NONE. At last, for every pointer (i.e., P_i) in pointer mask (i.e., P), the reading result is calculated as presented in the following equation:

$$\text{read}_{\text{num}} = \frac{\alpha_i - s_{\min}}{360 - \text{len}(\text{None})} \times \text{meter}_{\text{range}} \quad (16)$$

where read_{num} is the actual reading of the meter and $\text{meter}_{\text{range}}$ is the real range of the meter, which is the only prior knowledge our method needs.

IV. EXPERIMENT DETAILS AND SETTINGS

The experiments in this study are conducted using Ubuntu 20.04.3 and Python 3.8. We use PyTorch 1.10.3 with CUDA 11.3 on a single NVIDIA GeForce RTX 3090. The experimental process and implementation details are described below.

A. Motion Blur Simulation

Accurately simulating motion blur is a prerequisite for ensuring the robustness of our method. In our cases, the motion blur is mainly generated by the relative movement of the camera. Thus, the motion blur can be well represented in the following equation:

$$g = r \left(\frac{1}{T} \int_{t=0}^T f(t) dt \right) \quad (17)$$

where T denotes the exposure time of camera, $f(t)$ represents the pixel value captured by the camera at time t , and $r(\cdot)$ is the nonlinear camera response function (CRF) following [44], which can be formulated as in the following equation:

$$r(x) = x^{\frac{1}{2.2}}. \quad (18)$$

Therefore, a pixel of the motion blurred image is approximated by the weighted accumulation of its surrounding pixels along the moving direction of the camera. Furthermore, the accumulation process can be seen as a convolution process as presented in the following equation:

$$\frac{1}{T} \int_{t=0}^T f(t) dt \approx K * f \quad (19)$$

where K denotes the motion blur kernel. Specifically, we first use the fspecial function in MATLAB¹ to simulate linear motion in arbitrary directions and generate a linear blur kernel. In the following, we generate more diverse motion blur kernels using rotation matrices. Thus, K can be formulated as below

$$K = R(\theta) \cdot B(k, \phi) \quad (20)$$

where $B(k, \phi)$ denotes linear motion blur with kernel size of k , and direction with ϕ . And $R(\theta)$ denotes the rotation matrix with a rotation angle of θ . In our case, ϕ varies between 10° and 80° , while θ varies between 0° and 180° .

B. Dataset Preparation

Due to the scarcity of pointer meter datasets for public industrial scenarios, the experiment dataset is mainly derived from two publicly available datasets on the web and PMI dataset² [26]. We summarize the collected data and simulate the motion blur by applying enormous motion blur kernel to the data. Finally, the data are divided into two datasets: Motion blurred detection (MB-Detection) and motion blurred segmentation (MB-Segmentation), aiming to train and evaluate Yolov5s with MB-MixUp and FD-U-net, respectively.

1) *MB-Detection*: It is made from the first publicly available object detection dataset on the web,³ and real industrial scene images from PMI dataset. Then the random direction motion blur kernel with size of 36~44 is added into test set. In total, there are 1804 images in training set and 842 images in test set of 35 types of meters, with image size of 640×640 .

¹<https://www.mathworks.com/help/images/ref/fspecial.html>

²https://github.com/zzfan3/electric_meter_detect_recognize

³Pointer meter object detection dataset on the web.

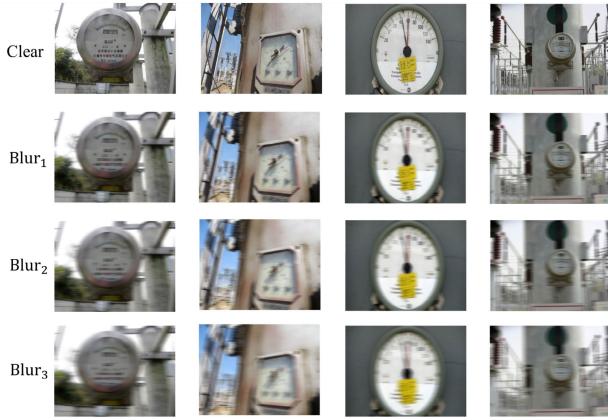


Fig. 12. Examples of motion-blurred images in Blur₁, Blur₂, and Blur₃.

2) *MB-Segmentation*: Seven types of 140 meters images are collected from the second web open-source dataset,⁴ and 28 types of 560-m images are selected from PMI dataset. Moreover, the meter is manually read to label the real readings. Next, the images are resized to the size of 448 × 448. For training set: 500 each of clear images, motion-blurred images with a kernel size of 20~28, and the segmentation masks are prepared. As for the test set: 200 each of the three level motion-blurred images are set up with kernel sizes of 20~28, 28~36 and 36~44, respectively. And we denote them as Blur₁, Blur₂, and Blur₃, respectively. The examples of the three-level blur test sets are presented in Fig. 12.

C. Evaluation Metric

The evaluation metrics in this study primarily focuses on two aspects: direct meter reading accuracy and the ability to recover information from blurred images. We also evaluate our motion blur image judgment mechanism using classification accuracy and examine the performance of object detection.

1) *Image Quality Evaluation*: Peak signal to noise ratio (PSNR) [45], structural similarity index measure (SSIM) [46], universal quality index (UQI) [47], and multiscale structural similarity index measure (MSSIM) [48] are adopted to evaluate the effectiveness of our method in recovering blurred images. A higher PSNR score indicates better quality of the restored image. The SSIM and MSSIM scores range from 0 to 1, while the UQI score ranges from -1 to 1. In all three metrics, values closer to 1 indicate better quality of the deblurred image.

2) *Reading Value Evaluation*: The evaluation criteria for pointer meter readings conform to two error ratios: the relative error, which measures the error ratio of the pointer reading to the true reading, and the reference error, which measures the error ratio of the estimated pointer rotation angle to the true rotation angle. These error ratios are denoted as Error_{rel} and Error_{ref}, respectively, as shown in (21) and (22), where N denotes the total number of the reading samples.

Error_{rel}: This metric directly evaluates the error ratio of the reading value obtained by the method of this study and the

true labeled reading value, as in the following equation:

$$\text{Error}_{\text{rel}} = \frac{\sum_{i=1}^N \frac{|v'_i - V_i|}{V_i}}{N} \times 100\% \quad (21)$$

where v'_i denotes the reading value of the i th pointer meter and V_i denotes the true value of the i th pointer meter.

Error_{ref}: It is a metric to evaluate the error ratio between the rotation angle obtained from reading methods and the true rotation angle of the pointer as presented in the following equation:

$$\text{Error}_{\text{ref}} = \frac{\sum_{i=1}^N \frac{|\alpha'_i - A_i|}{A_i}}{N} \times 100\% \quad (22)$$

where α'_i denotes the reading rotation angle and A_i denotes the true rotation angle of the i th pointer meter.

3) *Object Detection Evaluation*: We evaluate our method by comparing it to recent object detectors in terms of precision (P), recall (R), mean average precision (mAP₅₀ and mAP_{50:95}), inference time (ms), and frames/s (FPS).

D. Implementation and Experiment Details

In this section, we describe the experimental process and provide details of the model training and results collection. The Yolov5-s model is trained on MB-Detection for 80 epochs using a batch size of 16 and a learning rate of 0.01. Data augmentation is applied, including MB-MixUp with a probability of 40% per image. The performance of DCPD-Laplace is evaluated for motion-blurred image judgment using a range of no-reference thresholds. Our FD-U-net model is trained in two stages. First, we pretrain the U-net discriminator using half of the sharp pointer meter segmentation training set, consisting of 448 × 448 images and masks, for 80 epochs with a learning rate of 0.001. Second, we jointly train the U-net discriminator and deblurring module using paired blur-sharp images from the MB-Segmentation dataset for the remaining 600 epochs. The training uses a batch size of eight, a learning rate of 0.0001, and decreases the learning rate by a factor of 0.98 once after ten epochs. Finally, JRA performs an out-of-range pointer check and completes the meter reading based on the segmentation masks. After these steps, we calculate the Error_{rel} and Error_{ref} for all of our test samples.

V. EXPERIMENT RESULTS

A. Meter Detection Results Using MB-MixUp

To demonstrate the effectiveness of MB-MixUp, recent robust object detection networks are evaluated and compared on the MB-Detection dataset. Specifically, we conduct experiments on the Yolov5 series [49], Yolox series [50], and Yolov5-s with MB-MixUp. The performance of each method is presented in Table I. Yolov5-s with MB-MixUp achieves optimal performance on mAP₅₀ at 99.1%, mAP_{50:95} at 91.6%, and shares the fastest inference time with Yolov5-s, as MB-MixUp does not increase the time consumption during forward inference. While Yolox-l and Yolox-x achieve the best precision and recall, respectively, they are much slower due to their deeper networks. Therefore, our experiments confirm that MB-MixUp is effective in maintaining meter detection accuracy under motion blur.

⁴Pointer meter segmentation dataset on the web.

TABLE I
COMPARISON OF DETECTION RESULTS OF DIFFERENT NETWORKS WITH YOLOV5-S ENHANCED BY MB-MIXUP ON MB-DETECTION

Method	P(%)	R(%)	mAP ₅₀ (%)	mAP _{50:95} (%)	Time(ms)
Yolov5-m	96.4	95.1	97.8	88.8	3.9
Yolov5-l	96.8	92.6	97.2	88.3	6.3
Yolov5-x	96.5	96.9	98.0	89.6	10.2
Yolox-s	98.1	98.5	98.1	88.0	2.8
Yolox-m	99.0	99.4	99.1	89.1	4.9
Yolox-l	99.2	99.5	98.8	90.3	8.8
Yolox-x	99.3	99.0	98.6	89.3	12.6
Yolov5-s	95.4	94.4	97.7	87.1	2.4
Yolov5-s+MB-MixUp	99.2	95.7	99.1	91.6	2.4

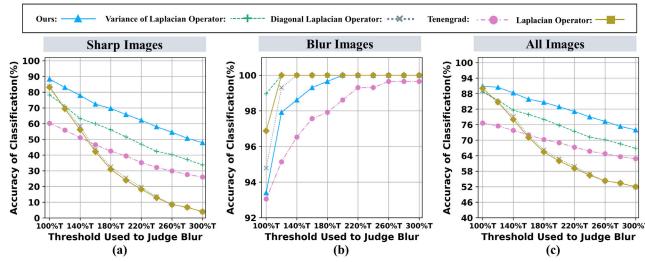


Fig. 13. Image type judgment accuracy using different blur degree measurement operators. T denotes the threshold used to judge blur. T is generated by the average value of the top 10% sharpest image in blur image under each specific operator. Comparison of classification accuracy in (a) sharp images, (b) motion-blurred images, and (c) all images.

TABLE II
AVERAGE CLASSIFICATION ACCURACY USING THRESHOLDS FROM 100% T TO 300% T

Method	Motion blur(%)	Sharp(%)	All(%)
Variance of Laplace [37]	99.65	57.21	78.43
Diagonal Laplace [38]	99.46	33.22	66.34
Tenengrad [39]	97.85	40.59	69.22
Laplace [35]	99.71	32.37	66.04
DCPD-Laplace (Ours)	97.7	70.36	84.02

B. DCPD-Laplace Motion Blur Judgment

No-reference blur judgment methods usually require a predefined threshold. We compare the Laplace, variance of Laplace, Diagonal Laplace, and Tenengrad methods with DCPD-Laplace on 1200 images, including 600 motion-blurred and 600 clear images. However, the variance values of different no-reference blur judgment methods may not fall into the same or similar value distribution, which means each method needs a specific threshold. To establish a standard, we set the initial threshold (T) for each method as the average value of the top 10% variance scores in blur images. As shown in Fig. 13, we increase T by 20% until reaching 3 T to examine the judgment accuracy of each method. Table II presents the overall accuracy for each method from T to 3 T . DCPD-Laplace shows an increase of 13.15% in judgment accuracy for sharp images and an increase of 5.59% in overall images compared to the suboptimal method [37]. The improvement is mainly due to the removal of mist-like noise in the background, which is usually harmless for semantic segmentation.

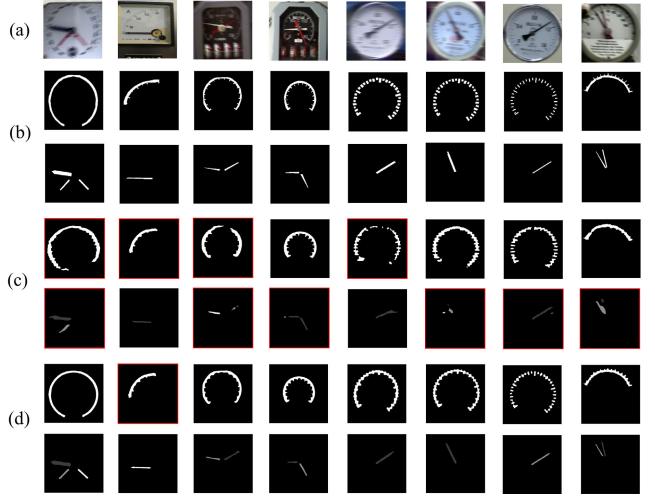


Fig. 14. Quantitative comparison of segmentation results with FD-U-net and U-net under motion blur. Segmentation results with red border denote the failure cases. (a) Motion-blurred images of multiple types of meters. (b) Ground truth segmentation masks. (c) Segmentation results of U-net. (d) Segmentation results of FD-U-net.

C. Restored Image Quality With FD-U-Net

To demonstrate that our FD-U-net can guarantee segmentation accuracy under extreme motion blur from the perspective of restored image quality, we extract the intermediate feature map after the deblurring module of FD-U-net. We then transform the feature map into a real image and compare it with the results of the latest motion deblurring networks, including MIMO-Unet [29], MIMO-Unet++ [29], DeepRFT [30], MPRnet [31], Uformer [32], and DeblurGANv2 [33].

As depicted in Table III, our comparison experiments are performed on three levels of blur test sets (Blur_1 , Blur_2 , Blur_3), and “NONE” denotes the motion-blurred images. The results indicate that even though the parameters of the deblurring module in FD-U-net are partially updated with the backpropagation of the U-net discriminator segmentation loss, it still maintains good image recovery performance. In the test set Blur_1 , our deblurring module achieves optimal performance on the SSIM, PSNR, and UQI indicators. In the test set Blur_2 , PSNR, SSIM, and MSSIM are optimal. In the test set Blur_3 , PSNR, and UQI are optimal, while SSIM and MSSIM perform the second best. Overall, the strategy of emphasizing the high-resolution deblurring stage seems promising in our FD-U-net. These results confirm that the proposed method is effective in balancing the segmentation and deblurring tasks.

D. Reading Precision on the Motion Blurred Image

As illustrated in Fig. 14, the conventional segmentation network, like U-net, produces abnormal segmentation errors due to pixel shift under motion blur on our motion blur datasets. The segmentation mask with a red border represents a segmentation failure case that can lead to severe reading errors. In particular, the segmentation error of the pointer can result in reading results with low accuracy. Fortunately, our proposed method can ensure much more accurate segmentation of the pointer and scale under extreme motion blur compared to [19],

TABLE III
COMPARISON OF DEBLURRING PERFORMANCE BETWEEN OUR FD-U-NET AND RECENT STATE-OF-THE-ART NETWORKS

Method	Blur ₁				Blur ₂				Blur ₃			
	SSIM	PSNR	MSSIM	UQI	SSIM	PSNR	MSSIM	UQI	SSIM	PSNR	MSSIM	UQI
NONE	0.716	21.48	0.706	0.960	0.681	20.46	0.664	0.953	0.660	19.72	0.638	0.944
MIMO-Unet [29]	0.920	30.13	0.926	0.993	0.844	26.24	0.851	0.987	0.782	24.41	0.786	0.983
MIMO-Unet++ [29]	0.924	30.96	0.929	0.993	0.841	26.21	0.847	0.986	0.783	24.57	0.787	0.981
DeepRFT [30]	0.927	30.93	0.931	0.991	0.868	26.90	0.875	0.986	0.825	24.67	0.810	0.983
MPRNet [31]	0.864	27.69	0.866	0.985	0.803	24.91	0.805	0.982	0.746	23.43	0.744	0.977
Uformer [32]	0.896	28.02	0.902	0.990	0.841	25.85	0.847	0.988	0.791	24.339	0.795	0.984
DeblurGANv2 [33]	0.855	25.44	0.865	0.976	0.801	23.80	0.809	0.975	0.763	22.71	0.769	0.970
FD-U-net (Ours)	0.932	31.28	0.928	0.994	0.871	26.74	0.877	0.985	0.819	24.80	0.798	0.984

TABLE IV
READING RESULTS OF DIFFERENT METHODS ON THE MOTION BLURRED POINTER METER DATASET

Method	Blur ₁		Blur ₂		Blur ₃	
	Error _{rel} (%)	Error _{ref} (%)	Error _{rel} (%)	Error _{ref} (%)	Error _{rel} (%)	Error _{ref} (%)
Fan et.al [26]	9.05	3.39	11.21	4.25	11.39	4.29
Lin et.al [20]	6.41	2.14	9.62	3.35	11.28	3.99
Chao et.al [19]	9.86	4.05	11.15	5.08	11.75	5.27
DeblurGANv2 [33] + Fan et.al [26]	2.83	0.98	2.92	1.01	3.35	1.17
DeblurGANv2 [33] + Lin et.al [20]	3.14	0.94	3.52	1.05	3.62	1.08
DeblurGANv2 [33] + Chao et.al [19]	6.34	2.35	7.24	2.66	8.06	3.00
DeepRFT [30] + Fan et.al [26]	2.77	0.95	2.85	0.97	3.18	1.11
DeepRFT [30] + Lin et.al [20]	3.23	0.97	3.31	0.99	3.48	1.04
DeepRFT [30] + Chao et.al [19]	5.26	1.96	7.27	2.70	7.34	2.76
Uformer [32] + Fan et.al [26]	2.80	0.99	3.04	1.10	6.63	3.51
Uformer [32]+ Lin et.al [20]	3.03	0.88	3.06	0.89	3.19	0.93
Uformer [32]+ Chao et.al [19]	6.71	2.46	7.24	2.71	9.72	4.12
FD-U-net + JRA (Ours)	1.54	0.48	1.59	0.53	1.67	0.60

TABLE V
COMPARISON OF 20 READING EXAMPLES OF DIFFERENT METHOD COMBINATIONS ON THE MOTION BLURRED POINTER METER DATASET

Meter serial	Actual reading (MPa, mA, °C)	Ours		[26]+Uformer		[20]+DeepRFT		[20]+Uformer		[26]+DeepRFT	
		Reading	Error(%)	Reading	Error(%)	Reading	Error(%)	Reading	Error(%)	Reading	Error(%)
1	1.100	1.099	0.09	1.048	4.73	1.051	4.45	1.082	1.64	1.079	1.91
2	0.620	0.622	0.32	0.623	0.48	0.618	0.32	0.615	0.81	0.615	0.81
3	6.300	6.047	4.02	6.160	2.22	6.181	1.89	6.010	4.60	6.030	4.29
4	0.610	0.612	0.33	0.613	0.49	0.613	0.49	0.607	0.49	0.606	0.66
5	6.100	5.957	2.34	6.176	1.25	6.189	1.46	5.885	3.52	5.901	3.26
6	0.630	0.617	2.06	0.571	9.37	0.582	7.62	0.589	6.51	0.595	5.56
7	1.100	1.104	0.36	1.119	1.73	1.122	2.00	1.101	0.09	1.100	0.00
8	0.610	0.607	0.49	0.612	0.33	0.616	0.98	0.605	0.82	0.606	0.66
9	6.300	6.069	3.67	6.335	0.56	6.335	0.56	6.062	3.78	6.064	3.75
10	0.610	0.61	0.00	0.636	4.26	0.625	2.46	0.615	0.82	0.610	0.00
11	1.090	1.09	0.00	1.112	2.02	1.105	1.38	1.088	0.18	1.089	0.09
12	0.640	0.615	3.91	0.598	6.56	0.616	3.75	0.598	6.56	0.597	6.72
13	0.600	0.616	2.67	0.646	7.67	0.646	7.67	0.625	4.17	0.625	4.17
14	6.400	6.116	4.44	6.422	0.34	6.422	0.34	6.057	5.36	6.065	5.23
15	6.300	6.054	3.90	5.899	6.37	5.863	6.94	5.928	5.90	5.938	5.75
16	0.610	0.607	0.49	0.640	4.92	0.637	4.43	0.613	0.49	0.609	0.16
17	6.300	6.018	4.48	6.343	0.68	6.332	0.51	6.003	4.71	5.977	5.13
18	6.400	6.136	4.13	6.324	1.19	6.285	1.80	6.039	5.64	6.025	5.86
19	1.110	1.111	0.09	1.102	0.72	1.105	0.45	1.106	0.36	1.105	0.45
20	6.300	6.162	2.19	6.478	2.83	6.505	3.25	6.147	2.43	6.154	2.32
Average	—	—	2.00	—	2.93	—	2.64	—	2.94	—	2.84

[20], and [26]. The methods employed in [19] are based on conventional computer vision, while [20] and [26] are based on deep learning. To ensure the fairness of the experiments, we will also compare the accuracy of the readings of [19], [20], and [26] after preprocessing the motion-blurred images using deblurring networks [30], [32], [33].

Table IV shows that the Error_{rel} of JRA associated with our proposed FD-U-net is only 1.54%, 1.59%, and 1.67% on the three levels of motion-blurred test sets, respectively. In con-

trast, Zheng et al. [19], Zuo et al. [20], and Fan et al. [26] obtain Error_{rel} and Error_{ref} that are almost 4~7 and 4~9 times as large as our method, respectively. Even when deblurring models like DeepRFT are added to other methods, JRA with FD-U-net still performs best and achieves a reduction of 1.23%, 1.26%, and 1.51% in Error_{rel}, compared to [26] strengthened by DeepRFT.

To analyze the advantages of our method over deep learning methods, we present 20 reading examples in Table V,

TABLE VI
RESULTS OF OUT-OF-RANGE POINTER CHECK

Example	Type 1	Type 2	Type 3	Type 4
Normal	14/15	15/15	13/15	14/15
Out-of-range	5/5	5/5	5/5	5/5
Accuracy(%)	95%	100%	90%	95%

TABLE VII

COMPARISON OF PERFORMANCE WHEN FD-U-NET UTILIZES DIFFERENT DBS

Indicator	8×Res	20×Res	8×FFT-Res	8×HFRA
PSNR	28.99	29.40	30.03	31.28
UQI	0.988	0.990	0.992	0.994
MSSIM	0.888	0.907	0.914	0.928
SSIM	0.901	0.910	0.912	0.922
Error _{rel} (%)	2.37	2.17	1.71	1.54
Error _{ref} (%)	0.87	1.01	0.64	0.48

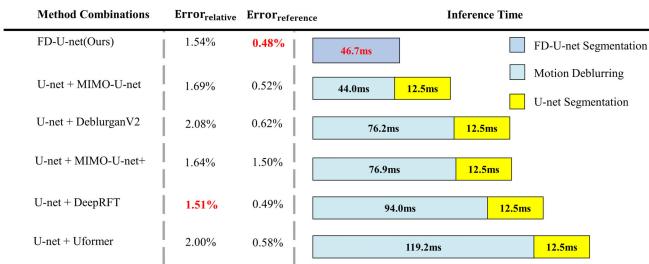


Fig. 15. Comparison of inference time and reading accuracy between FD-U-net and other two networks pipelines, using JRA as the reading algorithm.

comparing our method with other deep learning methods assisted by deblurring models, namely Uformer and DeepRFT. The “Error (%)” column represents the Error_{rel} of the meter reading. The Error_{rel} of all test samples with our method are below 5%, while the average Error_{rel} is 0.64% lower than the suboptimal method. The advantage of our method mainly comes from the robustness of JRA, since the reading process of JRA is robust to the segmentation error around the scale circle. As illustrated in Table VI, the out-of-range pointer check is examined on four types of meters, where out-of-range pointer accounts for 25% in 20 examples for each type. The out-of-range pointer is accurately detected, whereas a small number of normal pointers are judged as out-of-range, mainly due to the pointer approaching the start and end lines of the scales.

E. Ablation Study

The ablation experiments are conducted to analyze the benefits of the proposed components individually. Specifically, we illustrate the benefit of HFRA by comparing different residual blocks of the original resolution decoder stage in FD-U-net. As shown in Table VII, the restored intermediate feature map of HFRA has the optimal quality, with a PSNR of 31.28 dB and SSIM of 0.928. Moreover, the better quality of the feature map contributes to the best reading accuracy, with Error_{rel} and Error_{ref} of only 1.54% and 0.48%, respectively.

TABLE VIII
COMPARISON OF READING ACCURACY AND FPS WHEN FD-U-NET EMPLOYS DIFFERENT DEVICES AND INPUT RESOLUTIONS

Device	Resolution	Error _{rel} (%)	Error _{ref} (%)	FPS
NVIDIA GeForce RTX 3090 24GB GDR6X	336 × 336	1.97	0.64	28.50
	448 × 448	1.54	0.48	21.41
	560 × 560	1.49	0.48	14.92
	672 × 672	1.76	0.57	10.66
NVIDIA GeForce GTX 1080 Ti 11GB GDR5X	336 × 336	1.97	0.64	14.87
	448 × 448	1.54	0.48	8.93
	560 × 560	1.49	0.48	5.69
	672 × 672	1.76	0.57	4.24

We also compare our FD-U-net with two network combinations that first deblur and then segment, as presented in Fig. 15. FD-U-net has similar reading accuracy performance compared to U-net assisted by DeepRFT, while FD-U-net is 2.82 times faster, with an average inference time of 46.7 ms.

F. Hardware and Inference Speed Analysis

In real-world applications, it is crucial to find the trade-off between inference speed and reading accuracy. We conduct inference experiments with different input resolutions and GPU devices, with a batch size of one. As illustrated in Table VIII, both the Error_{rel} and Error_{ref} first decrease and then increase as the input resolution increases. Specifically, the Error_{rel} decreases from 1.54% to 1.49% when the input resolution varies from 448 × 448 to 560 × 560, but the inference speed decreases from 28.50 to 21.41 FPS. Therefore, we choose the resolution of 448 × 448 as the input of FD-U-net. Additionally, limited computational resources also need to be taken into account. While the NVIDIA GeForce RTX 3090 is a high-end GPU, the more cost-effective NVIDIA GeForce GTX 1080 Ti still enables FD-U-net to run at an inference speed of 8.93 FPS with the input resolution of 448 × 448.

VI. CONCLUSION

In this study, novel solutions for the detection and recognition of meter under motion blur is proposed. For meter detection, the developed MB-MixUp significantly improves the robustness of the object detector in motion-blur scenarios. Subsequently, the DCPD-Laplace enables the inspection devices to assert if the captured image is motion-blurred, which increases efficiency in practical use. For meter recognition and reading, the one-stage FD-U-net is capable of completing the feature segmentation of the pointer meter under extreme motion blur. Finally, the JRA algorithm is proposed to determine whether the pointer is out-of-range and robustly complete multiple types of meters’ reading based on the segmentation mask. The entire pipeline of our methods achieve 1.54% and 0.48% of the relative and reference error, respectively, in a tough environment, where the meter is far from the camera with motion blur interference and complex background.

Our future research will optimize the proposed algorithm. As revealed by the experiments to verify the quality difference between the intermediate feature map and the original map,

our HFRA structure is capable of retaining the corresponding features between the intermediate and original maps significantly even with the segmentation loss guidance. This inspires us to add an object detection branch in our future work and thereby realize the deblurring network for instance segmentation under multitasking learning. Additionally, we will study more challenging problems related to pointer meter readings, such as correction of deflection and dealing with large obstacle occlusion, to make the algorithm more robust.

REFERENCES

- [1] S. Lu, Y. Zhang, and J. Su, "Mobile robot for power substation inspection: A survey," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 4, pp. 830–847, Jan. 2017.
- [2] H. Jiale, L. En, T. Bingjie, and L. Ming, "Reading recognition method of analog measuring instruments based on improved Hough transform," in *Proc. IEEE 10th Int. Conf. Electron. Meas. Instrum.*, vol. 3, Aug. 2011, pp. 337–340.
- [3] L. Kang, H. Li, H. Zheng, and X. Li, "Research on automatic recognition algorithm of pointer meter based on machine vision," in *Proc. 6th Int. Conf. Intell. Comput. Signal Process. (ICSP)*, Apr. 2021, pp. 265–268.
- [4] L. Huang, X. Ma, F. Fang, and B. Zhou, "Device target checking for power patrol robot based on objectness estimation," in *Proc. 15th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Nov. 2020, pp. 73–78.
- [5] X. Zhang, X. Dang, Q. Lv, and S. Liu, "A pointer meter recognition algorithm based on deep learning," in *Proc. 3rd Int. Conf. Adv. Electron. Mater., Comput. Softw. Eng. (AEMCSE)*, Apr. 2020, pp. 283–287.
- [6] Y. Li and B. Jia, "Research on reading recognition of pointer meter based on improved U-net network," *Proc. SPIE*, vol. 12257, pp. 170–179, Mar. 2022.
- [7] D. Zhou, Y. Ying, Z. Jie, and W. Ku, "Intelligent reading recognition method of a pointer meter based on deep learning in a real environment," *Meas. Sci. Technol.*, vol. 33, no. 5, 2021, Art. no. 055021.
- [8] Y. Zheng, M. Chen, J. Peng, and D. Qi, "An automatic reading method for pointer meter based on one-stage detector," *Proc. SPIE*, vol. 11584, pp. 42–48, Aug. 2020.
- [9] Y. Fang, Y. Dai, G. He, and D. Qi, "A mask RCNN based automatic reading method for pointer meter," in *Proc. Chin. Control Conf. (CCC)*, Jul. 2019, pp. 8466–8471.
- [10] Y. Rao, F. Yan, Y. Zhuang, and G. He, "A real-time auto-recognition method for pointer-meter under uneven illumination," in *Proc. IEEE Int. Conf. Real-Time Comput. Robot. (RCAR)*, Aug. 2019, pp. 890–894.
- [11] C. Zhu, C. Zhang, X. An, Z. Jia, Y. Xu, and H. Liu, "Automatic indication recognition of dual pointer meter in thermo-hygrometer calibrator," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 3013–3017.
- [12] R. Zhao, D. Liu, F. Qu, and X. Zhang, "Motion blurred image restoration based on flight control system," in *Proc. 37th Chin. Control Conf. (CCC)*, Jul. 2018, pp. 9771–9775.
- [13] Z. Chao, Y. Yao, and Y. Han, "Reading pointer meter through one stage end-to-end deep regression," in *Proc. Chin. Conf. Pattern Recognit. Comput. Vis. (PRCV)*, vol. 13022, Nov. 2021, pp. 348–360.
- [14] E. Ullah, R. Nawaz, and J. Iqbal, "Single image haze removal using improved dark channel prior," in *Proc. 5th Int. Conf. Model., Identificat. Control (ICMIC)*, Aug. 2013, pp. 245–248.
- [15] X. Meng, C. Lv, F. Cai, H. Liu, J. Wang, and M. Shuai, "Research on reading recognition method of pointer meter based on magnifying the virtual circle," in *Proc. Int. Conf. Mach. Learn. Knowl. Eng. (MLKE)*, Feb. 2022, pp. 191–194.
- [16] M. Hung and C. Hsieh, "Automatic pointer meter reading based on machine vision," in *Proc. IEEE 4th Int. Conf. Image, Vis. Comput. (ICIVC)*, Jul. 2019, pp. 32–35.
- [17] H. Gao, M. Yi, J. Yu, J. Li, and X. Yu, "Character segmentation-based coarse-fine approach for automobile dashboard detection," *IEEE Trans. Ind. Informat.*, vol. 15, no. 10, pp. 5413–5424, Oct. 2019.
- [18] Q. Lv, Y. Rao, S. Zeng, C. Huang, and Z. Cheng, "Small-scale robust digital recognition of meters under unstable and complex conditions," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–13, 2022.
- [19] C. Zheng, S. Wang, Y. Zhang, P. Zhang, and Y. Zhao, "A robust and automatic recognition system of analog instruments in power system by using computer vision," *Measurement*, vol. 92, pp. 413–420, Oct. 2016.
- [20] L. Zuo, P. He, C. Zhang, and Z. Zhang, "A robust approach to reading recognition of pointer meters based on improved mask-RCNN," *Neurocomputing*, vol. 388, pp. 90–101, May 2020.
- [21] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 816–832.
- [22] B. Howells, J. Charles, and R. Cipolla, "Real-time analogue gauge transcription on mobile phone," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 2369–2377.
- [23] J. Huang, J. Wang, Y. Tan, D. Wu, and Y. Cao, "An automatic analog instrument reading system using computer vision and inspection robot," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 9, pp. 6322–6335, Sep. 2020.
- [24] Z. Dong, Y. Gao, Y. Yan, and F. Chen, "Vector detection network: An application study on robots reading analog meters in the wild," *IEEE Trans. Artif. Intell.*, vol. 2, no. 5, pp. 394–403, Oct. 2021.
- [25] S. Han, F. Yang, H. Jiang, G. Yang, N. Zhang, and D. Wang, "A smart thermography camera and application in the diagnosis of electrical equipment," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–8, 2021.
- [26] Z. Fan, L. Shi, C. Xi, H. Wang, S. Wang, and G. Wu, "Real time power equipment meter recognition based on deep learning," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–15, 2022.
- [27] Z. Li, Y. Zhou, Q. Sheng, K. Chen, and J. Huang, "A high-robust automatic reading algorithm of pointer meters based on text detection," *Sensors*, vol. 20, no. 20, p. 5946, Oct. 2020.
- [28] Y. Gong, Y. Li, and H. Zhu, "Restoration algorithm of blurred UAV aerial image based on generative adversarial network," in *Proc. 40th Chin. Control Conf. (CCC)*, Jul. 2021, pp. 7201–7206.
- [29] S. Cho, S. Ji, J. Hong, S. Jung, and S. Ko, "Rethinking coarse-to-fine approach in single image deblurring," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4621–4630.
- [30] X. Mao, Y. Liu, F. Liu, Q. Li, W. Shen, and Y. Wang, "Intriguing findings of frequency selection for image deblurring," 2021, *arXiv:2111.11745*.
- [31] S. W. Zamir et al., "Multi-stage progressive image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14816–14826.
- [32] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu, and H. Li, "Uformer: A general U-shaped transformer for image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17662–17672.
- [33] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8877–8886.
- [34] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.
- [35] L. Zhou, W. Min, D. Lin, Q. Han, and R. Liu, "Detecting motion blurred vehicle logo in IoV using filter-DeblurGAN and VL-YOLO," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 3604–3614, Apr. 2020.
- [36] S. Zheng, Y. Wu, S. Jiang, C. Lu, and G. Gupta, "Deblur-YOLO: Real-time object detection with efficient blind motion deblurring," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.
- [37] P. Said, P. Domènec, and G. Miguel, "Analysis of focus measure operators in shape-from-focus," *Pattern Recognit.*, vol. 46, no. 10, p. 1016, 2012.
- [38] A. Thelen, S. Frey, S. Hirsch, and P. Hering, "Improvements in shape-from-focus for holographic reconstructions with regard to focus operators, neighborhood-size, and height value interpolation," *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 151–157, Jan. 2009.
- [39] A. S. Malik and T.-S. Choi, "A novel algorithm for estimation of depth map using image focus for 3D shape recovery in the presence of noise," *Pattern Recognit.*, vol. 41, no. 7, pp. 2200–2225, Jul. 2008.
- [40] Y. Gao, L. Gao, X. Li, and X. V. Wang, "A multilevel information fusion-based deep learning method for vision-based defect recognition," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 3980–3991, Jul. 2020.
- [41] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, p. 294–310.
- [42] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, vol. 9351, Oct. 2015, pp. 234–241.
- [43] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

- [44] Y. Tai et al., "Nonlinear camera response functions and image deblurring: Theoretical analysis and practice," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2498–2512, Oct. 2013.
- [45] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 2366–2369.
- [46] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [47] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Process. Lett.*, vol. 9, no. 3, pp. 81–84, Mar. 2002.
- [48] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2003, pp. 1398–1402.
- [49] G. Jocher et al. *YOLOv5*. Accessed: Sep. 2022. [Online]. Available: <https://github.com/ultralytics/yolov5>
- [50] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.



Hongyu Zhang is currently pursuing the B.S. degree in software engineering with the University of Electronic Science and Technology of China, Chengdu, China. And, he plans to continue his academic pursuits by pursuing the M.S. and Ph.D. degrees in computer science.

His research interests include industrial vision, medical image processing, and remote sensing object detection.



Yunbo Rao (Member, IEEE) received the B.S. degree from Sichuan Normal University, Chengdu, China, in 2003, and the M.E. and Ph.D. degrees from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, in 2006 and 2012, respectively.

From 2009 to 2011, he was a Visiting Scholar of electrical engineering with the University of Washington, Seattle, WA, USA. Since 2012, he has been with the School of Information and Software Engineering, University of Electronic Science and Technology of China, where he is currently an Associate Professor. His research interests include image segmentation, 3-D reconstruction, video enhancement, and medical image processing.



Jie Shao (Member, IEEE) received the B.E. degree in computer science from Southeast University, Nanjing, China, in 2004, and the Ph.D. degree in computer science from The University of Queensland, Brisbane, QLD, Australia, in 2009.

He was a Research Fellow with The University of Melbourne, Melbourne, VIC, Australia, from 2008 to 2011, and the National University of Singapore, Singapore, from 2012 to 2014. He is currently a Professor with the Center for Future Media, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include spatial databases and multimedia information retrieval.



Fanman Meng (Member, IEEE) received the Ph.D. degree in signal and information processing from the University of Electronic Science and Technology of China, Chengdu, China, in 2014.

From 2013 to 2014, he was a Research Assistant with the Division of Visual and Interactive Computing, Nanyang Technological University, Singapore. He is currently a Professor with the School of Information and Communication Engineering, University of Electronic Science and Technology of China. He has authored or coauthored numerous technical articles in well-known international journals and conferences. His current research interests include image segmentation and object detection.

Dr. Meng is a member of the IEEE Circuits and Systems Society. He was a recipient of the Best Student Paper Honorable Mention Award at the 12th Asian Conference on Computer Vision, Singapore, in 2014, and the Top 10% Paper Award at the IEEE International Conference on Image Processing, Paris, France, in 2014.



Jiansu Pu received the Ph.D. degree from The Hong Kong University of Science and Technology, Hong Kong, in 2013.

He is currently an Associate Professor with the Department of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include information visualization, visual analysis in spatiotemporal data, time series, and social networks.