European Journal of Interdisciplinary Research and Development
Volume-19 September 2023
Website: www.ejird.journalspark.org ISSN (E): 2720-5746

# INTELLIGENT MOTION BLUR SYSTEM FOR SOCCER PLAYERS

Nada Thanoon Ahmed 1,
Maysaa Hameed Abdulameer 2,
Abbas Akram Khorsheed 1
Nadia Mahmood Hussien 1,
Yasmin Makki Mohialden 1
1Department of Computer Science, College of Science,
Mustansiriyah University, Baghdad, Iraq
2Iraqi Commission for Computers and Informations,
Informatics Institute for Postgraduate Studies Baghdad, Iraq

**Abstract**:
Motion blur is a very important part of how we see moving things. This effect shows up as a visible trail along the path of the object. It is caused by the way that film and digital cameras handle relative motion and light integration. In this paper, we developed a software method to create motion blur on the YasminadNadia 25th Arabian Gulf Cup Iraq dataset, which has the faces of Arabian Gulf Cup Iraq players and is made public on the websites Kaggle and GitHub. We also found the best kernel motion blur size for each image in the dataset. The proposed method was made using the openCV, NumPy, and pillow libraries.

**Keywords**: Intelligent motion, video Captureing, Blur Image, Gulf Cup Dataset, open CV.

## Introduction

People with money, technology, and experience are no longer the only ones capable of producing films. Designers, illustrators, and artists that work with new media can create movies using digital filmmaking on a desktop computer. This sparked ground-breaking research. Because of short films, music videos, animation, new media, and graphics, digital filmmaking is evolving. [1]. Blur occurs when the camera and the object both move at the same time. Motion blur can be seen with an angle-length point-spread function. These parameters must be well calculated for blind motion-blurred image restoration. Without Gaussian noise, there are numerous methods for determining the direction and length of motion blur in an image. These motion blur parameters can then be used by a nonblind deconvolution algorithm. Photos with blur can reveal movement. [2, 3].

Training on simulated cases blurs motion. Our variable blur model performs well with a variety of data and makes few assumptions about the photographs it utilizes. Finally, we can say: (1) For deblurring finger vein images, an adversarial network created an Inception-Resnet-v2 backbone-based DeblurGAN-v2 network. (2) A vein feature-guided finger vein image inpainting network can repair damaged images. The network corrects the main image and the original finger vein in two phases. (3) A selection of finger vein image restoration tasks based on adaptable deep reinforcement learning has been proposed for the first time. This system guides information about vein features to the optimal restoration tasks by using a reward function and constraints on vein features.

The second section of a research article discusses similar works, the third section discusses the proposed approach, the fourth section discusses the dataset, the fifth section discusses experimental results, and the sixth section discusses the conclusion and what else could be done in the future.

## 1. RELATED WORKS

In [2017], they advocate for a data-driven approach.The latent image must be modeled in order to obtain a prior. Their strategy is centered on comprehending motion flow, which enables the model to determine the source of the blur regardless of image content. Although this is a simpler learning challenge, latent image priors are used iteratively. Using a fully convolutional deep neural network (FCNN), they directly predict motion flow from the blurred image and then recover the unblurred image from the estimated motion flow. Their FCN is the first to provide a universal mapping from hazy images to dense motion flow. They train the FCN by replicating motion flows rather than categorizing hazy images. Extensive testing on genuinely blurred images shows that the proposed technique outperforms the current standard [4].

[2018] This work describes a novel method for correcting for motion blur on deformable body images in which the blur (w) varies during the frame. For each image region, a subset-based approach estimates and reduces motion blur. According to synthetic and experimental data, the method improves DIC displacement accuracy [5].

[2019] They produce motion-blurred images by mixing two unblurred images that are successively combined. They motivate and create a new "line prediction" layer for a neural network architecture in order to train a system to regress from image pairs in motion-blurred lurred images that span the input image pair's capture time.Frame interpolation is used to build a huge synthetic dataset of motion-blurred images and their inputs for training this model, which requires a large amount of data. In addition, they generate a high-quality test set of real motion blur images from slow-motion videos in order to compare our model to a variety of baseline motion blur approaches. In terms of accuracy, our model outperforms baselines and is several orders of magnitude faster. [6].

[2020] This research provides a novel method for simulating captured motion by applying motion blur to a single image. With little human effort, filtering generates artificial motion. By appropriately addressing object boundaries, it avoids frequent filtering artifacts. They show how our system handles difficult situations, including multi-directional blur, reflections, multiple objects, and motion-related aesthetic affects. Our post-processing method allows for fine-grained control of motion blur without capturing it directly [7].

[2021] They proposed that a deep convolutional neural network be used to smooth noisy or fuzzy images taken quickly. The images are fused sequentially or simultaneously using the deblur LSTM and deblurMerger neural networks. Training is improved by gradient, adversarial, and spectral normalization. GOPRO is used to construct the training dataset from pairs of noisy or fuzzy images and the ground truth, a sharp image. They used both synthetic and real-world image pairs to train their networks. In both qualitative and quantitative aspects, the suggested method outperforms state-of-the-art methods. Although Deblur LSTM outperforms it, Deblur Merger outperforms it with less calculation time [8].

[2021] In this paper, they offer a new objective blur level (BL) measure based on a reference image using point spread function/blur kernel analysis. They discovered that our BL metric accurately represented the perceived image quality of motion-blurred images better than SSIM and PSNR in most circumstances. Furthermore, in low-light and low-texture images, where SSIM and PSNR are prone to failing to describe the blurriness or sharpness of the image [9], our technique outperforms SSIM and PSNR metrics.

[2022] They suggest using video to forecast the 3D motion, form, and appearance of extremely motion-blurred objects. They depict the fuzzy appearance of a fast-moving object by parametrizing its 3D location, rotation, velocity, acceleration, bounces, form, and texture during a certain time interval. Differentiable rendering reduces pixel-wise reprojection error to the input video by backpropagating through a rendering pipeline that accounts for motion blur by averaging the graphics output across short time intervals to estimate all parameters. The camera exposure gap time is calculated using the same optimization [10].

[2023] DRL-FVRestore, a deep reinforcement learning-based technique for choosing and restoring finger vein images, is proposed. They classified the restoration tasks as image denoising and augmentation, image deblurring, and image inpainting. They first propose considering the finger vein restoration task as a sequential decision-making process and training an agent to select the image restoration task adaptively based on the status of the finger vein image to gradually repair and improve its quality. DRL has recently been employed in image processing. Yu et al. were the first to use DRL to learn a technique for gradually recovering damaged images using a toolbox of relevant techniques. Their improved multi-path CNN can dynamically select pathways for different image regions, allowing for spatially shifting image denoising. Furuta et al. developed the first pixelated restoration framework, PixelRL. They presented the first DRL application to finger vein image restoration based on these findings. They also create a reward function based on vein feature limitations to help the trained agent acquire all vein feature information and choose the optimum restoration technique. [11] Appl. Sci. 2023, 13, 699, 5 of 33

## 2. PROPOSED METHOD

Motion blur is a type of blur that is used to give images a directed blur effect. Motion blurring has traditionally been related to photography and video capturing. When fast actions are captured in photographs or video recordings, motion blur is common. When recording a single frame, fast motions may cause the image to alter before the frame is finished. The proposed program makes use of motion blur to collect data for us.

When implementing image convolution, the size of the matrix kernel provided determines the filter intensity noticed in the resulting images. Larger kernels can provide the appearance of rapid motion, whereas smaller kernels produce the perception of slower motion. Here's an example of a 5-by-5 filter.

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} \left(\frac{1}{9}\right)$$

The following Python libraries are used by the software:

### 1. OpenCV

OpenCV is one of the greatest open-source computer vision and machine learning software libraries. It allows developers to create large-scale projects that include image processing, motion detection, and image segmentation, among other things. To enable real-time computer vision algorithms, OpenCV for Python combines the best of the OpenCV C++ API and the Python language [12,15].

**2. NumPy**

NumPy is a fork of the well-known Numeric Array object that adds two crucial components to create a realistic environment for scientific computing. A global object, sometimes known as a Ufunc object, and an N-dimensional array object are the components. The additional elements must be configured to work on top of any N-dimensional objects stored in a Ndarray. An array with dimensions can be defined using the two key bits of information contained inside it [13,16,17].

**3. Pillow**

The Python Imaging Library (PIL) is a popular image processing library. PIL can be used to show images, generate thumbnails, resize, rotate, convert between file formats, enhance contrast, filter, and perform other digital image processing techniques, among other things. PIL supports image formats such as PNG, JPEG, GIF, TIFF, and BMP. It also has strong image processing and graphics capabilities. The image module is one of the most significant classes in PIL. It has an in-built capability for performing tasks such as loading images, saving, changing image format, and creating new images [14]. Figure 1 depicts the activity diagram [18], which describes the process sequences for the software.

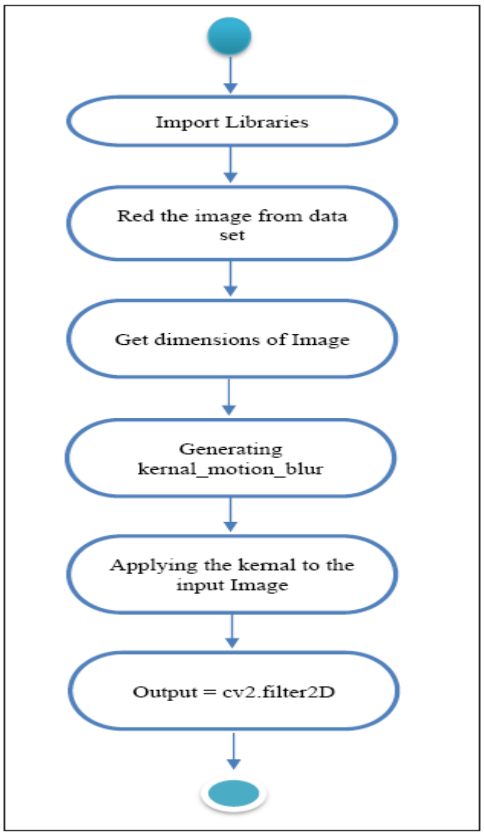



Figure 1. The activity diagram of process sequences for the software

European Journal of Interdisciplinary Research and Development
Volume-19 September 2023
Website: www.ejird.journalspark.org ISSN (E): 2720-5746

## 4. Dataset

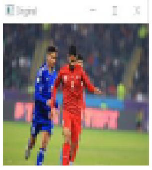
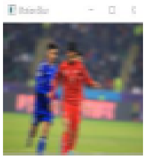
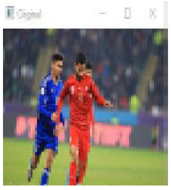
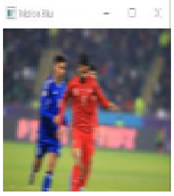
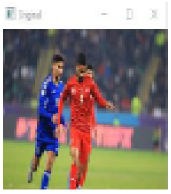
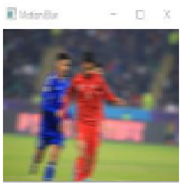
We used the YasminadNadia 25th Arabian Gulf Cup Iraq data collection, which contains faces for Arabian Gulf Cup Iraq payers and is freely available on the Kaggle (https://www.kaggle.com/datasets/yasminmakki/yasminadnadia-25th-arabian-gulf-cup-iraq) and GitHub Websites.

## 5. Experimental results

The suggested approach is applied to the dataset, one example of which is shown in detail below, and the optimal value for motion blue is selected. Table 1 displays the attributes of the original image and the motion-blurred image, as well as the kernel motion blur size. Table 2 shows the relationship between kernel motion blur size and pixel count in a blurred image. Figure 2 depicts the variation in kernel motion blur size for image 1.

Table 1. The characteristics of the original image and the motion-blurred image with the size of its kernel, motion blur

motion blur

| Image no | Original image | Kernel Motion Blur size | Motion blurred image | Image Dimension | Image Height | Image Width | Number of Channels |
|---|---|---|---|---|---|---|---|
| 1 |  | 5 |  | (168, 300, 3) | 168 | 300 | 3 |
| Colors In Motion Blur image | [((64, 64, 64), 25966), ((155, 175, 76), 11120), ((254, 153, 159), 3706), ((231, 57, 50), 3589), ((6, 20, 120), 1873), ((73, 152, 209), 1612), ((137, 151, 116), 531), ((184, 128, 81), 462), ((90, 23, 15), 398), ((181, 178, 197), 376), ((12, 18, 57), 274), ((21, 51, 215), 162), ((103, 87, 141), 115), ((163, 72, 103), 7), ((72, 113, 225), 2)] | | | | | | |
| Pixel Count of bluer image | 50193 | | | | | | |
| Colors In original image | [((80, 83, 90), 26605), ((155, 175, 76), 11318), ((253, 152, 156), 3448), ((255, 68, 59), 3296), ((13, 33, 168), 1777), ((75, 151, 203), 1078), ((77, 28, 11), 955), ((7, 14, 42), 663), ((137, 151, 116), 523), ((154, 24, 22), 204), ((8, 15, 87), 181), ((167, 113, 67), 161), ((255, 221, 198), 99), ((89, 105, 201), 51), ((222, 221, 255), 15), ((182, 60, 117), 14), ((31, 40, 241), 7), ((181, 130, 173), 2), ((58, 69, 26), 2), ((36, 175, 178), 1)] 50400 | | | | | | |
| Pixel Count of bluer image | 5400 | | | | | | |
| 1 |  | 10 |  | (168, 300, 3) | 168 | 300 | 3 |
| Colors In Motion Blur image | [((64, 64, 64), 25966), ((155, 175, 76), 11120), ((254, 153, 159), 3706), ((231, 57, 50), 3589), ((6, 20, 120), 1873), ((73, 152, 209), 1612), ((137, 151, 116), 531), ((184, 128, 81), 462), ((90, 23, 15), 398), ((181, 178, 197), 376), ((12, 18, 57), 274), ((21, 51, 215), 162), ((103, 87, 141), 115), ((163, 72, 103), 7), ((72, 113, 225), 2)] | | | | | | |
| Pixel Count of bluer image | 50100 | | | | | | |
| 1 |  | 15 |  | (168, 300, 3) | 168 | 300 | 3 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Colors In Motion Blur image | [((81, 84, 91), 27372), ((155, 175, 76), 10796), ((238, 141, 154), 3914), ((195, 51, 41), 3054), ((174, 123, 86), 1252), ((9, 19, 68), 1229), ((18, 41, 181), 852), ((72, 152, 210), 797), ((81, 90, 180), 163), ((80, 36, 24), 154), ((169, 165, 179), 82), ((119, 64, 108), 72), ((130, 185, 156), 29)] | | | | | | |
| Pixel Count of bluer image | 49766 | | | | | | |
| 1 |  | 20 |  | (168, 300, 3) | 168 | 300 | 3 |
| Colors In Motion Blur image | [((64, 64, 64), 26493), ((155, 175, 76), 10246), ((164, 58, 50), 3882), ((254, 154, 157), 3848), ((9, 20, 76), 1418), ((67, 145, 202), 1316), ((157, 151, 165), 646), ((29, 46, 159), 632), ((115, 125, 87), 408), ((127, 59, 91), 229), ((240, 52, 47), 160), ((203, 150, 99), 57)] | | | | | | |
| Pixel Count of bluer image | 49335 | | | | | | |
| 1 |  | 60 |  | (168, 300, 3) | 168 | 300 | 3 |
| Colors In Motion Blur image | [((64, 64, 64), 28782), ((155, 175, 78), 10681), ((255, 255, 255), 8713), ((166, 76, 84), 4951), ((153, 153, 153), 2235), ((173, 124, 71), 1861), ((78, 68, 140), 1637), ((70, 147, 201), 518), ((199, 53, 45), 18), ((89, 163, 128), 4)] | | | | | | |
| Pixel Count of bluer image | 59400 | | | | | | |

Table 2. Shows the relationship between the size of the kernel motion blur and the number of pixels in the blurred image in one image.

| Image no | Size | Pixel Count |
|---|---|---|
| 1 | 5 | 50193 |
| 1 | 10 | 50100 |
| 1 | 15 | 49766 |
| 1 | 20 | 49335 |
| 1 | 60 | 59400 |

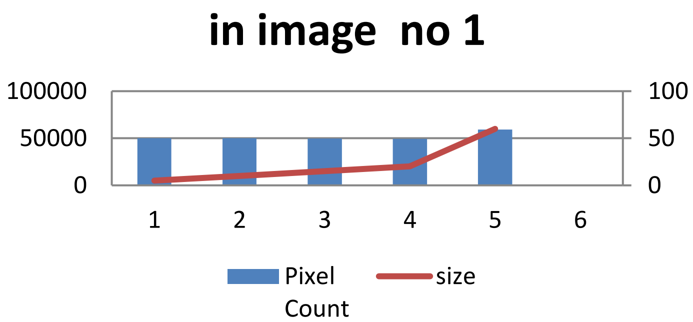



Figure 2. Depicts the variation in kernel motion blur size for image 1.

We determined that the ideal size for this image is 15 based on the experimental findings in Tables 1 and 2 and Figure 2.

## 3. CONCLUSION

We suggest a way to apply a motion blur effect to a data collection using Python libraries. The motion blur effect will make it look like you took the photo while moving in a certain direction. For example, you can make a picture look like it was taken from a moving car. The software can figure out the perfect kernel motion blur size for each image in the dataset. We can make a software proposal using the method for removing irregular motion blur from several blurry images.

## ACKNOWLEDGEMENTS

The Authors would like to thank Mustansiriyah University (https://uomustansiriyah.edu.iq/) Baghdad, Iraq for its support in the present work

## REFERENCES

[1] S. R. Walter and M. Hanson, Motion blur: Graphic moving imagemakers. Laurence King Publishing, 2004.

[2] S. Tiwari, V. P. Shukla, A. Singh, and S. Biradar, "Review of motion blur estimation techniques," Journal of Image Graphics, vol. 1, no. 4, pp. 176-184, 2013.

[3] S. Dai and Y. Wu, "Motion from blur," in 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1-8: IEEE.

[4] D. Gong et al., "From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2319-2328.

[5] A. Lavatelli, S. Turrisi, and E. Zappa, "A motion blur compensation algorithm for 2D DIC measurements of deformable bodies," Measurement Science Technology, vol. 30, no. 2, p. 025401, 2018.

[6] T. Brooks and J. T. Barron, "Learning to synthesize motion blur," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6840-6848.

[7] X. Luo, N. Z. Salamon, E. J. I. T. o. V. Eisemann, and C. Graphics, "Controllable motion-blur effects in still images," vol. 26, no. 7, pp. 2362-2372, 2018.

[8] M. Abdullah-Al-Mamun, V. Tyagi, and H. Zhao, "A New Full-Reference Image Quality Metric for Motion Blur Profile Characterization," IEEE Access, vol. 9, pp. 156361-156371, 2021.

[9] S. Zhang, A. Zhen, and R. L. Stevenson, "Deep motion blur removal using noisy/blurry image pairs," Journal of Electronic Imaging, vol. 30, no. 3, pp. 033022-033022, 2021.

[10] Z. Zhong, M. Cao, X. Ji, Y. Zheng, and I. Sato, "Blur Interpolation Transformer for Real-World Motion from Blur," arXiv preprint arXiv:.11423, 2022.

[11] R. Gao, H. Lu, A. Al-Azzawi, Y. Li, and C. Zhao, "DRL-FVRestore: An Adaptive Selection and Restoration Method for Finger Vein Images Based on Deep Reinforcement," Applied Sciences, vol. 13, no. 2, p. 699, 2023.

[12] A. F. Villán, Mastering OpenCV 4 with Python: a practical guide covering topics from image processing, augmented reality to deep learning with OpenCV 4 and Python 3.7. Packt Publishing Ltd, 2019.
[13] R. W. Kadhim, M. jaffar Raheem, Y. M. Mohialden, and N. M. Hussien, "A Review of the Implementation of NumPy and SciPy Packages in Science and Math," Turkish Journal of Computer Mathematics Education, vol. 13, no. 03, pp. 663-667, 2022.
[14] P. Umesh, "Image processing in python," CSI Communications, vol. 23, no. 2, 2012.
[15] Hussien, N. M., Hussain, S. A. K., Mohialden, Y. M., Joshi, K., & Shah, S. K. (2023, September). Real time image processing monitoring system based IoT. In AIP Conference Proceedings (Vol. 2771, No. 1). AIP Publishing.
[16] Ethar Abdul Wahhab Hachim, and Yasmin Makki Mohialden, "Cloud-based digital watermarking model for medical image integrity," Scientific Research Journal of Engineering and Computer Science, Vol-3, Issue 5, 2023
[17] Ali, J. J., Shati, N. M., & Gaata, M. T. (2020). Abnormal activity detection in surveillance video scenes. TELKOMNIKA (Telecommunication Computing Electronics and Control), 18(5), 2447-2453.
[18] Alsaad, S. N., & Hussien, N. M. (2021). IoT based Message Alert System for Emergency Situations. International Journal of Computing and Digital Systems, 10(1).