Eng Int Syst (2022) 4: 279–284

Engineering
Intelligent Systems

Fast Object Detection for Continuous Images in Line Inspection

Xiaoliang Zhang*, Jianjun Cheng, Xuefeng Bai and Hanyu Zhang

Lvliang power supply company of State Grid Shanxi Electric Power Company., Lvliang, Shanxi, 033000, People's Republic of China.

It is important to guarantee the stable and reliable operation of a power system to ensure power supply safety. The rapid expansion of emerging power systems has brought significant challenges relating to power line inspection, especially under hazardous conditions. The existing vision-based line inspection approach emerges as one promising solution. However, the required computation is prohibitive as it requires a convolutional neural network (CNN) inference for each image frame. In this work, we address this problem by investigating block matching and extrapolation algorithms. These two algorithms exploit the motion information in consecutive frames of real-time videos, thus avoiding the expensive CNN inference for every image frame. According to the experiment evaluation, the processing rate is drastically increased by introducing a very limited amount of computation that leverages the temporal pixel motion. Moreover, the precision loss is negligible when the window size is small while the rate of improvement is significant.

Keywords: Power line inspection, continuous vision, block matching, neural network

1. INTRODUCTION

The stable and reliable operation of a power system is a fundamental guarantee to ensure power supply safety. The growing dependence of contemporary societies on electrical energy imposes tremendous challenges on the monitoring, inspection, and preservation of electric-powered energy grids to ensure the uninterrupted supply of electricity. The transmission line is a critical component in the power supply system and the safety of power transmission lines is closely associated with the ordinary operation of the whole electricity delivery system. Power outages caused by the failures of power transmission lines are becoming more frequent with the rapidly expanding grid infrastructure.

Conventional approaches to power line inspection typically involve field surveys using manpower. However, the previous inspection methods are not realistic for modern power grids for two reasons: 1) the rapid expansion of power delivery systems significantly increases the human resources required for field inspections; 2) the wide area covered by power delivery systems, which includes harsh and complex environments, means engineers face hazardous working conditions, such as storms and hurricanes, which increases the risk level of power line inspections. Hence, there is a trend towards the use of unmanned aerial vehicles (UAVs) to assist with line inspection tasks, enhancing inspection efficiency and safety. While UAVs equipped with image acquisition modules improve the accessibility of power line inspection, the transmitted images mostly are manually analyzed by humans in the backend [1]. The emerging adoption of UAVs eases the difficulty of on-site inspection, but still requires a large amount of human work. As a result, overall system efficiency is still low.

To solve this problem, some computer vision (CV) pre-processing techniques are used to further improve inspection efficiency. The rapid development of deep learning algorithms ensures the human-level accuracy of object detection. By

*Address for correspondence: Xiaoliang Zhang, Lvliang power supply company of State Grid Shanxi Electric Power Company., Lvliang, Shanxi, 033000, People's Republic of China, Email: zhangxiaoliangei@126.com

analyzing continuous images transmitted by UAVs and robotics, the line inspection processing pipeline can be realized by introducing neural network models [2,3]. The authors in [4] exploit convolutional neural networks (CNNs) to analyze and screen the inspection contents through object detection. Similarly, CNN is also adopted to recognize various types of power line equipment with very strong robustness [5]. After the interesting contents of line inspection are extracted, the workload associated with human inspection is significantly reduced since most of the distracting and redundant content in the collected on-site image sequences have been removed.

However, some drawbacks still exist in the aforementioned works as they fail to take into consideration the feasibility of a CNN-aided vision system for line inspection. According to [6], the current CNN-based continuous vision system such as SSD [7] and Faster R-CNN [8] requires a computational complexity that is one order of magnitude higher than the popular mobile devices. Despite investigating some complexity reduction tricks [9–11], the pruned neural network comes at a significant performance loss and the accuracy is not guaranteed.

In this work, we focus on improving the continuous vision system that is applied to detect power line anomalies, enabling new schemes to be realized in real use cases. Instead of increasing computing capability or reducing complexity, we propose a novel motion estimation algorithm to discover the motion information inherent in consecutive frames of real-time videos. This avoids the need to perform an expensive CNN inference on every frame, hence the processing rate is drastically increased by introducing a limited amount of computation that leverages temporal pixel motion.

The remainder of this paper is organized in the following form. Section 2 presents the proposed fast scheme to improve the processing rate of object detection for power line inspection. The evaluation and corresponding experiments are conducted and detailed in Section 3 to evaluate the performance of our proposed algorithms. Finally, Section 4 concludes the paper.

2. PROPOSED RATE IMPROVEMENT SCHEME FOR POWER LINE INSPECTION

2.1 Computation of Convolutional Neural Networks

In this section, we first briefly introduce the inherent computation of existing CNN models. Then the computational complexity of the key operation, convolution, is analyzed. Throughout this paper, we consider deep CNN models to be composed of several functional layers in the feed-forward pattern. The commonly used functional layers include the convolution layer, the fully connected (FC) layer, and the pooling layer.

Each FC layer computes the linear combination of a given input vector $\mathbf{x}$ with the weights matrix $\mathbf{W}$. The computation of FC layers can be realized through matrix-vector multiplication. The convolution layer is the most important component in the feature extraction task. For each input feature map that has several image channels, the CONV layer performs a two-dimensional (2-D) convolution operation on the given input feature map and associated filters in the shape of $K \times K$. The hidden features within the input images are extracted by the convolution layer by performing 2D convolution on the input feature map $\mathbf{I}$ with shape $C \times H \times W$ using the M trained $K \times K$ filters $\mathbf{W}$. The yield output of the convolution layer, named the output feature map $\mathbf{O}$, with shape $M \times E \times F$ can be described as the following computation process (See Figure 1):

$$\mathbf{O}[m, e, f] = \sum_{c=1}^{C} \sum_{i=1}^{K} \sum_{j=1}^{K} \mathbf{I}[c, e \times S + i, f \times S + j] \times \mathbf{W}[m, c, i, j].$$

where $\mathbf{W} \in \mathbb{R}^{M \times C \times K \times K}$ is the weight tensor composed of M trained filters. It should be noted that each filter contains one C-channel kernel in the shape of $K \times K$ while $f(\cdot)$ represents the activation function. Based on different tasks, the popular activation functions include the rectified linear unit (ReLU) $f(x) = max(x, 0)$ and the sigmoid activation $f(x) = (1 + exp^{-x})^{-1}$.

The 2D convolution is a type of computation-intensive arithmetic operation, which requires significant multiply-accumulate (MAC) operations. According to the analysis the Figure 1, the computation complexity caused by convolution takes up over 90% of the overall CNN inference. Taking Figure 1 as the example, for a convolution layer which receives an input feature map $\mathbf{I}$ with the shape $C \times H \times W$ and produces output that has the shape $M \times E \times F$, the required number of MAC operations is given by:

$$\text{MAC}_{\text{CONV}} = 2 \cdot E \cdot F \cdot K^2 \cdot M \cdot C,$$

where K^2 MAC operations are needed for each channel of each filter.

For a continuous video with a frame rate of R, assuming that we process each incoming frame independently, a total of R frames of images should pass the CNN model and the inference should be performed R times per second. As a result, the total computation complexity per second is as follows:

$$\text{MAC}_{\text{Total}} = \sum_{i=1}^{L} 2 \cdot R \cdot E_i \cdot F_i \cdot K_i^2 \cdot M_i \cdot C_i,$$

where L is the number of convolution layers. For simplicity, here we only consider convolution complexity since convolution operation contributes to the majority of the overall complexity.

The complexity introduced by CNN inference is prohibitive compared to the limited computing power of traditional computer architecture. One idea is to reduce the required arithmetic complexity by using fast convolution algorithms, such as Winograd and fast Fourier transform (FFT) [12]. Although fast algorithms can accelerate the inference speed of CNN given the same computing resources, the speedup effect remains insignificant when the frame rate of videos is high. In the following sections, we introduce our proposed scheme to improve the processing speed of CNN-based power line inspection.

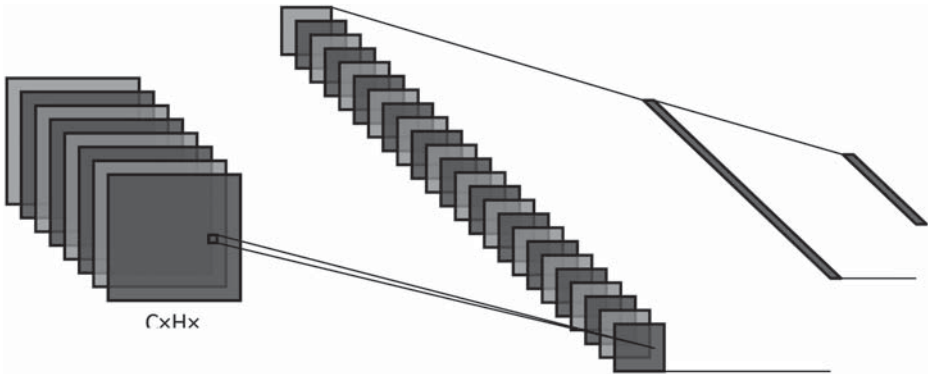

Figure 1 Illustration of convolution computation in CNN.

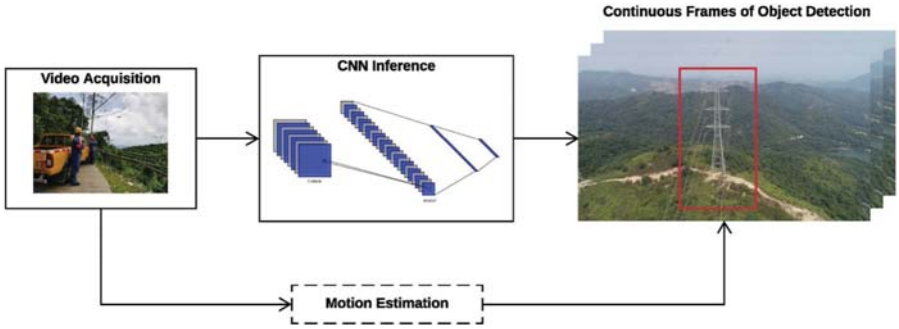

Figure 2 The proposed processing pipelines for power line inspection.

2.2 Motion Estimation for CNN

In this section, we present the details of our proposed motion estimation algorithm that improves the continuous object detection rates for power line inspection. To effectively solve the contradiction between computational complexity and limited computation resources, we utilize two techniques, block matching (BM) and ROI extrapolation, to enhance the processing rate by adding a very limited computation workload. The proposed system design increases the practicability and efficiency of the continuous vision system that is responsible for power line inspection. The proposed processing pipelines for power line inspection are illustrated in Figure 2. The proposed design comprises several parts, which are explained in detail in the following sections.

(1) Block Matching

Motion estimation is a popular image processing algorithm that estimates how the collected pixels are moving between consecutive image frames. Motion estimation is one of the most crucial algorithms applied to image processing since video denoising and stabilization both require motion information. Temporal denoising uses pixel motion information to remove noisy pixels by replacing the noise-free data in previous frames. Moreover, up-sampling is able to increase the frame rate through intersecting interpolating frames be- tween successive real frames. Block matching [13] is the most widely used motion estimation algorithm since it achieves a good trade-off between accuracy and efficiency.

The basic calculation of BM is conducted in the following steps. Due to the space limitation, we only present the main ideas here. We refer interested readers to [13] for more details.

First, BM divides an image frame into $L \times L$ macroblocks. For each MB, the algorithm searches for the closest matched one in the previous frames by adopting the sum of absolute differences (SAD) as the matching metric for all L^2 macroblocks. The search process is performed within a 2-D window with a total of $(2d + 1) \times (2d + 1)$ pixels, where d is the search range.

The key factor in different BM search strategies is to trade-off between matching accuracy and computational complexity. The traditional accurate approach is to perform an exhaustive search, which requires a large amount of computation with a complexity of $L^2 \times (2d + 1)^2$ operations for each macroblock. The other BM search schemes improve computation efficiency by reducing the arithmetic complexity at the cost of a slight loss in accuracy. The three-step search (TSS) searches only a small range of the search window by decreasing d, thus reducing the number of required operations to $L^2 \times (1 + 8 \log 2(d + 1))$ for each macroblock.

Finally, the BM algorithm obtains the motion vector (MV) for each macroblock that defines the pixel offset between one specific macroblock and its closest matched block in the previous frames. The MV can be treated as the estimation of each macroblock's motion. Specifically, an MV $< u, v >$ of a macroblock at coordinate $< x, y >$ denotes that the macroblock moves from the coordinate $< x + u, y + v >$ in the previous frame. The required storage memory space for MVs is low since the MVs can be efficiently encoded and stored. In this case, a total of

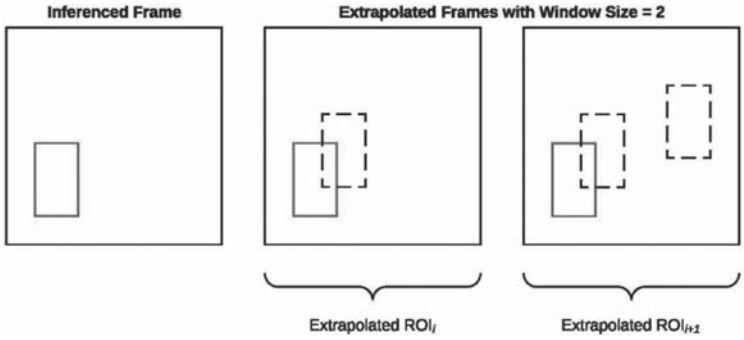

Figure 3 The system architectures of the proposed sensor node.

$\lceil \log_2(2d+1) \rceil$ bits are enough to preserve the needed information.

(2) ROI Extrapolation

Typical vision applications, object tracking and object detection, involve localizing and classifying multiple objects within an image by bounding the objects with boxes, named the region of interest (ROI). CNN models are widely used and dominate the tracking and detection tasks since CNNs have powerful feature extraction capabilities. However, performing an inference for every incoming image frame requires a huge amount of computing resources and is unrealistic for real implementations.

Based on [6], we propose an ROI extrapolation scheme to improve the processing rate of object detection for power line inspection. The essential idea is to exploit the temporal motion information and predict the ROI while avoiding unnecessary CNN inferences. Large parts of ROI are estimated and predicted by BM and ROI extrapolation algorithms. Compared to the expensive CNN inference, BM and ROI extrapolation require far less computation. Therefore, given the same computing capability, our proposed scheme can significantly improve the processing rate.

The ROI extrapolation aims at accurately estimating ROIs using the MVs for a specific frame without performing CNN inference. The first step is to calculate the average motion vector μ for a ROI according to vector v_i as shown in the following Eq. (4):

$$\mu_F = \sum_{i=1}^{N} \frac{v_i}{N},$$

where N is the total number of pixels inside the ROI, and v_i represents the MV of the i-th pixel within the bound.

The BM algorithm introduces noise to the MVs, making it unable to find the idle matched block, the noise is evaluated using the following SAD-related metric:

$$\alpha_F^i = 1 - \frac{\mathrm{SAD}_F^i}{255 \times L^2},$$

where SAD_F represents the SAD value of the i-th macroblock in frame F. The SAD-related metric is finally normalized into the range of [0,1].

The SAD-related metric is useful to filter the introduced noise of BM using the following equation, which is calculated in a moving average manner:

$$MV_F = \beta \times \mu_F + (1-\beta) \times MV_{F-1},$$

The location associated with the new ROI is updated by combining the previous frame and the obtained MV_F as follows:

$$R_F = R_{F-1} + MV_{F-1}.$$

The ROI extrapolation that utilizes the pixel-level temporal motion information greatly simplifies the complexity of continuous vision tasks. Recalling Eq. (3), the required total complexity with ROI extrapolation is:

$$\mathrm{MAC}_{\mathrm{extra}} = W \cdot L^2 \cdot (2d+1)^2 + \sum_{i=1}^{L} 2 \cdot \frac{R}{W} \cdot E_i \cdot F_i \cdot K_i^2 \cdot M_i \cdot C_i,$$

where N denotes the number of convolution layers while W is the window size of extrapolation. In this case, the costly CNN inference is reduced by a factor of W.

3. EVALUATION RESULTS

Based on the proposed algorithms, we conduct a detailed evaluation. In this section, the experiment parameters are presented in the case of power line inspection. Our proposed scheme is evaluated using the YOLO algorithm which employs CNNs [14]. The results are also compared with other methods to demonstrate the effectiveness of the proposed algorithms.

3.1 Experiment Setup

(1) Simulation Platform

The detailed specifications for the simulation and experiments are given as follows. The baseline YOLO model is implemented on the deep learning platform PyTorch [15]. We use the pre-trained YOLO which is available on PyTorch for a fair comparison. The inference of YOLO [14] is performed on one Nvidia Titan X GPU equipped with the Intel Xeon-E5 1660 CPU. For simplicity, the motion estimation algorithm is implemented using the Python language. Therefore, all the proposed schemes are run and tested on software.

To fully exploit the computing power of the GPU card and the error redundancy of the CNN model, the YOLO net is quantized with a INT8 data format, which gains near

Table 1 The used parameters for motion estimation.

Parameter	Value
Window size	$W = 2, 4, 8, 16$
Macroblock number	$L = 8$
Search range	$d = 8$

Table 2 Comparison of average precision using different augmentation algorithms and IoU thresholds.

Algorithms	IoU Thres = 0.1	IoU Thres = 0.3	IoU Thres = 0.5
Original	72.2%	69.2%	55.4%
Ex-ROI-W = 2	71.5%	67.7%	50.1%
Ex-ROI-W = 4	70.4%	65.2%	47.2%
Ex-ROI-W = 8	70.1%	57.6%	41.5%
Ex-ROI-W = 16	68.7%	51.1%	37.2%

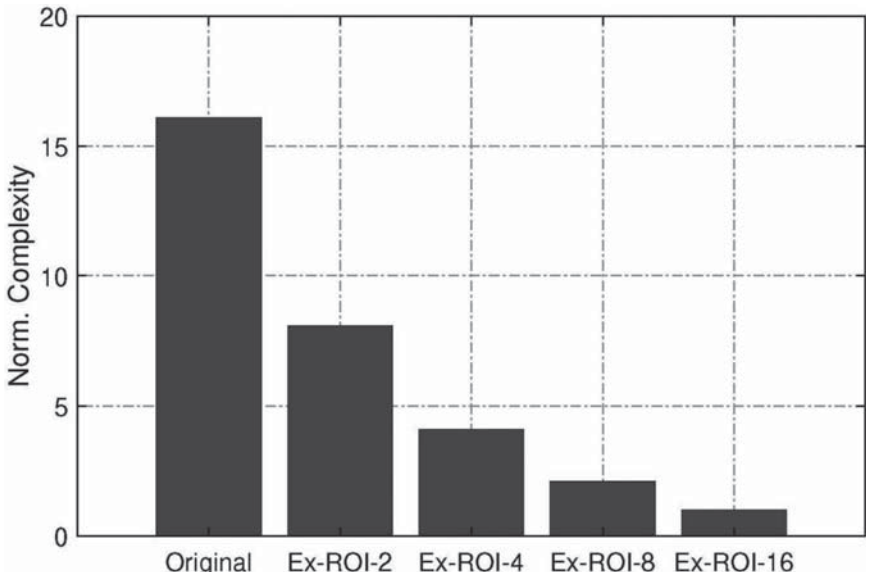

Figure 4 Computational complexity for different ROI extrapolation schemes and the original YOLO model.

4× speedup compared to the FLOAT32 quantization. The introduced accuracy degradation is negligible. The input of the tested model is composed of cropped video sequences with a resolution of 960 × 960. The video is captured by the camera on a DJI Mavic 2 drone. The original high-resolution video is resized into a low-resolution version since a higher resolution requires more computing resources.

(2) Parameter Selection

For the block matching and ROI extrapolation algorithms, there are several key parameters to be determined. These parameters include the number of macroblocks L, the extrapolation windows size W, and the search range d.

These parameters are selected empirically and we also refer to the parameter selection in [6]. Table 1 summarizes the used parameters for motion estimation. Four extrapolation window sizes, $W = 2, 4, 8, 16$ are chosen to achieve a balance between accuracy and the processing rate improvement. The number of macroblocks is set to $L = 8$ while the search range is set to $d = 8$, based on the typical values in [6].

3.2 Experiment Results

The experiment results are given in this section to evaluate the proposed algorithms. A comparison is also given in Table 2. First, the standard Intersect-over-Union (IoU) metric is used as the accuracy metric for object detection. IoU denotes the ratio between intersection and union areas between the predicted ROI and the correct results. One detection is considered as the true positive if the IoU value exceeds the threshold.

First, we test the degradation loss introduced by the BM and ROI extrapolation schemes. The performance comparison using various augmentation methods is shown in Table 2. The results for four different window sizes $W = 2, 4, 8, 16$ are shown. The ROI extrapolation leads to some degree of average precision loss. But the degradation is acceptable when W <= 8, considering the processing frame rate is improved by a factor of W. The results show that the proposed ROI extrapolation method is necessary in the case of continuous vision-based power line inspection.

We also conduct a comparison regarding computation complexity with the original CNN model, as shown in Figure 4. Since ROI extrapolation avoids the expensive CNN

inference, large amounts of MAC computation are saved. As a result, nearly linear speedup is achieved, which means that the process frame rate is also improved by around a factor of W. However, more significant acceleration comes at the cost of higher accuracy loss.

4. CONCLUSION

In this paper, we aim at improving the processing rate of a continuous vision system that is applied to detect the power line inspection task, making the new schemes realizable in real use cases. Instead of increasing computing capability or reducing complexity, we propose a novel motion estimation algorithm to discover the motion information inherent in consecutive frames of real-time videos. This avoids the need to perform expensive CNN inference on every frame, hence the processing rate is drastically increased by introducing a very limited amount of computation which leverages the temporal pixel motion. The experiment results show that the precision loss is negligible when the window size is small while the rate of improvement is significant given the same computing resources.

ACKNOWLEDGEMENT

This work is sponsored by the Lvliang power supply company of State Grid Shanxi Electric Power Company under grant SGSXLL00FCJS2000224. The project name is Realization of Panoramic View of On-site Safety Management and Control based on VR Technology.

REFERENCES

1. H. Zhao, R. Dai, and C. Xiao, "A machine vision system for stacked substrates counting with a robust stripe detection algorithm," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.
2. X. Tao, D. Zhang, Z. Wang, X. Liu, H. Zhang, and D. Xu, "Detection of power line insulator defects using aerial images analyzed with convolutional neural networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.
3. Z. Zhao, G. Xu, Y. Qi, N. Liu, and T. Zhang, "Multi-patch deep features for power line insulator status classification from aerial images," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 3187–3194.
4. R. Jenssen, D. Roverso et al., "Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning," *International Journal of Electrical Power & Energy Systems*, vol. 99, pp. 107–120, 2018.
5. X. Xiang, N. Lv, X. Guo, S. Wang, and A. El Saddik, "Engineering vehicles detection based on modified faster R-CNN for power grid surveillance," *Sensors*, vol. 18, no. 7, p. 2258, 2018.
6. Y. Zhu, A. Samajdar, M. Mattina, and P. Whatmough, "Euphrates: Algorithm-soc co-design for low-power mobile continuous vision," in *ACM/IEEE 45th Annual Inter- national Symposium on Computer Architecture (ISCA)*. IEEE, 2018, pp. 547–560.
7. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European Conference on Computer Vision*. Springer, 2016, pp. 21–37.
8. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object de- tection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.
9. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
10. S. Lin, R. Ji, C. Yan, B. Zhang, L. Cao, Q. Ye, F. Huang, and D. Doermann, "Towards optimal structured CNN pruning via generative adversarial learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2790– 2799.
11. J.-H. Luo, H. Zhang, H.-Y. Zhou, C.-W. Xie, J. Wu, and W. Lin, "ThiNet: pruning CNN filters for a thinner net," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 10, pp. 2525–2538, 2018.
12. A. Lavin and S. Gray, "Fast algorithms for convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4013–4021.
13. M. Jakubowski and G. Pastuszak, "Block-based motion estimation algorithms—a sur- vey," *Opto-Electronics Review*, vol. 21, no. 1, pp. 86–102, 2013.
14. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
15. A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison,
16. L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in NIPS-W, 2017