

Automatic Recognition of Concrete Cracks in Industrial and Civil Buildings Based on CNN + SVM/Random Forest

Zhiheng Yang *

Leeds College, Southwest Jiaotong University, Chengdu, Sichuan, 611756, China

* Corresponding author Email: a147258369@my.swjtu.edu.cn

Abstract: Concrete cracks are a key indicator for evaluating the safety of industrial and building structures Civil buildings. Traditional manual detection methods have low efficiency, strong subjectivity, and high labor costs, which cannot meet the needs of large-scale automated monitoring Significant progress has been made in learning based crack detection algorithms, with most existing models For the development of transportation infrastructure (bridges, pavements, tunnels), due to significant differences, the performance will decrease by 15% to 25% when directly migrated to building scenarios In terms of lighting, background texture, and crack morphology. In addition, currently over 95% of research Only focusing on binary classification, unable to provide crack width and Severity is crucial for engineering practice. Although pixel level segmentation methods can Implementing crack quantification relies on extremely expensive manual labeling (every 10-20 minutes) Images make large-scale applications impractical. To address these challenges, this article proposes an automatic crack recognition scheme for lightweight concrete used in industrial and civil buildings, combined with CNN feature extraction based on SVM/random forest classifier. This scheme utilizes CNN Superior generalization of hierarchical feature extraction and SVM/random forest in small samples Plot. The four pre trained CNN backbone networks (VGG16, ResNet50, MobileNetV3, EfficientNet Lite) are System comparison and introduction of ZCA whitening and anti distillation to enhance Has robustness to complex interferences such as shadows and stains. For edge deployment (Smartphones, drones, Raspberry Pi), full frame lightweighting, and model compression Implemented support vector pruning and parameter quantization. It is crucial to have coarse particles By mining the severity of cracks, innovative classification of crack severity was achieved without pixel level segmentation The implicit mapping between CNN features and crack width greatly reduces annotation costs. Experiments on public datasets (SDNET2018, Crack500) and self built industrial/civilian datasets Constructing a crack dataset shows that this approach increased F1 scores by 6% in complex situations Interference scene. The optimized MobileNetV3+compressed SVM model only achieves 92 KB, single image inference time of 0.8 ms on Raspberry Pi 4B, crack severity classification The accuracy rate is 87%. These results fully meet the requirements of practical engineering, Efficient and deployable solutions for building health assessment.

Keywords: Concrete Crack Recognition; CNN+SVM/Random Forest; Lightweight Deployment; Crack Severity Classification; Edge Computing.

1. Introduction

Concrete is the most widely used structural material in modern civil engineering. In the long-term service life of industrial and civil buildings, cracks are inevitably generated under the combined effects of structural loads, temperature changes, chemical corrosion, and foundation settlement. Cracks are not only surface defects that affect aesthetics, but also [1, 2]

Accumulated structural damage. If not detected in a timely manner, corrosive media will penetrate along cracks, leading to steel corrosion and reduced structural bearing capacity. Therefore, accurate crack detection and severity assessment are crucial for ensuring building safety.

For decades, crack detection has mainly relied on manual visual inspection using tools such as magnifying glasses and crack width gauges [3]. This method has inherent limitations: extremely low efficiency (a team of 5-8 inspectors takes 3-5 days to complete a medium-sized residential community), highly subjective results among inspectors, high operational risks for high-level inspections, and high long-term costs [4]. These shortcomings urgently require intelligent automated detection technology.

With the development of deep learning, intelligent crack

detection based on CNN has become a research hotspot [5]. Unlike traditional image processing methods that rely on manual feature engineering, convolutional neural networks automatically learn multi-level features from labeled images [6]. However, there are three core issues that constrain the engineering applications of industrial and civil buildings in existing research.

Firstly, the scene is severely mismatched. Most models are trained on transportation infrastructure datasets (bridges, roads, tunnels) that have relatively simple backgrounds and continuous cracks [7]. In contrast, architectural scenes have complex backgrounds (painted walls, tiles, metal components) and intricate, discontinuous cracks that are easily obscured by shadows, stains, and graffiti. The model trained on the traffic dataset showed a 15% -25% decrease in F1 score when applied to buildings [8].

The second is that the detection task is shallow. Currently, over 95% of research only focuses on binary classification and cannot achieve quantitative assessment of crack severity [9]. In engineering practice, the width of cracks determines the maintenance strategy. According to GB 50204-2015, cracks with a width < 0.1mm are very minor and only require monitoring, while cracks with a width ≥ 0.1 mm need to be repaired in a timely manner [10]. The binary model cannot

provide this key information.

Thirdly, the quantification cost is high. The existing severity quantification methods rely on pixel level segmentation and require manual labeling of each crack pixel (10-20 minutes per image) [11]. For a large number of inspection images of existing buildings, this method is not economically feasible.

In addition, model deployment is a key bottleneck. High precision CNN models have huge parameters and can only run on GPU servers [12]. However, inspection equipment in practice includes smartphones, drones, and handheld terminals with limited computing power.

To address these issues, this paper proposes an intelligent crack recognition framework for industrial and civil buildings based on CNN+SVM/random forest. The main innovative points are: (1) Enhanced robustness by combining ZCA whitening and anti distillation features; (2) Lightweight design of the entire framework, convenient for edge deployment; (3) Non segmentation based crack severity classification using only global image labels; (4) Comparative Study of SVM and Random Forest Severity Classification.

2. Related Work

2.1. Deep Learning-Based Concrete Crack Detection

CNN is widely used in crack detection due to its powerful feature extraction ability. Foreign scholars have pioneered this field. In 2016, Cha et al. first applied deep CNN to crack detection, achieving an accuracy of over 98% on 40000 images [1]. In 2017, Zhang et al. introduced Use VGG16 pre trained on ImageNet for transfer learning [2]. In 2018, Zou et al. proposed DeepCrack, which achieves pixel level segmentation through multi-scale fusion [3] Recent research has focused on enhancing robustness and lightweight deployment. Yamaguchi et al. proposed a CrackAttentionNet that integrates channel and spatial attention mechanisms [7]. Huang et al. introduced adversarial domain adaptation to improve cross scene generalization ability [8]. Liu et al. designed a lightweight network using depthwise separable convolution, reducing parameters by 90% [9] Domestic research is developing rapidly. In 2018, Wang et al. applied Faster R-CNN to bridge crack detection [6]. In 2020, Wu et al. proposed CrackGAN for data augmentation [12]. Recent work Fan et al. implemented detection based on MobileNetV2 on Android smartphones [10], while Wang et al. developed a detection system based on drones [11].

2.2. CNN + Traditional Classifier Fusion Models

The end-to-end CNN model requires a large amount of training dataset, which is typically not available for industrial and civil buildings [13]. Traditional classifiers such as SVM and random forest have excellent generalization ability in small sample scenarios [14]. Therefore, the fusion of CNN+traditional classifiers has become very important for small sample crack detection [15].

Simonyan et al. validated that CNN deep features using SVM perform better than fine tuned CNN in small sample tasks [16]. Donahue et al. proposed the DeCAF framework and confirmed the effectiveness of CNN features [17]. In crack detection, Kumar et al. showed that in the case of samples < 500 [4], the F1 score of CNN+SVM was 5% -8% higher than that of CNN fine-tuning. Park et al. compared

CNN, CNN+SVM, and CNN+Random Forest and found that the fusion model is more robust to noise [5] Domestic scholars have also explored fusion models. Ke et al. applied CNN+SVM to bridge crack detection using ResNet50 features [13]. Yang et al. combined VGG16 features with SVM for complex backgrounds [14]. Niu et al. optimized the SVM kernel function for tunnel scenes [15]. Luo et al. and Chen et al. respectively applied CNN+random forest to crack detection in tunnels and bridges [18,19]

2.3. Summary of Existing Deficiencies

There are three core shortcomings in current research: (1) mismatched scenarios: lack of specialized models for cracks in industrial and civil buildings; (2) High quantification cost: relying on pixel level segmentation; (3) Shallow Fusion Research: Lack of Feature Robustness and System Optimization of Models Lightweight.

3. Methodology

The proposed framework consists of three core modules: feature robustness enhancement, lightweight deployment, and crack severity classification. Input as concrete surface image; The output includes binary detection (crack/non crack) and severity classification (mild/obvious).

3.1. Feature Robustness Enhancement Module

The images of cracks in building scenes are subject to various interferences, Shadows, stains, and complex backgrounds [20]. This module enhances feature robustness through three steps.

3.1.1. Multi-Backbone Feature Extraction

We selected four typical CNN models (VGG16, ResNet50, MobileNetV3, EfficientNet Lite) as feature extractors. VGG16 and ResNet50 are typical deep models with strong feature expression;

MobileNetV3 and EfficientNet Lite are lightweight models for edge devices [21-24]

3.1.2. Dual-Strategy Feature Enhancement

We introduced ZCA whitening and adversarial distillation to enhance features.

ZCA whitening eliminates feature correlation and normalizes the feature distribution [25]. For high-dimensional CNN features, ZCA transforms the feature matrix to make the covariance matrix an identity matrix, reducing sensitivity to changes in lighting and shadow interference.

Adversarial distillation shifts robust feature expression from teacher models to student models Model [8]. We use ResNet50 as the teacher and MobileNetV3/EfficientNet Lite as the lightweight CNN student. Students learn to adapt to training labels through adversarial training while minimizing the distance between their feature distribution and that of the teacher.

3.1.3. Adaptive SVM Parameter Optimization

The performance of SVM depends on hyperparameters, especially the penalty coefficient C and kernel Parameter γ [4]. We propose adaptive optimization based on validation set error types: (1) if the false positive rate is high, increase C to reduce overfitting to noise; (2) If the false positive rate is high, adjust γ to Enhance the identification of fine cracks; (3) Iterate until the F1 score reaches its maximum.

3.2. Lightweight Deployment Module

To meet the needs of edge devices, we have designed a

lightweight deployment module that includes backbone network selection, classifier compression, and inference optimization.

3.2.1. Lightweight Backbone Selection

Based on feature extraction comparison, we choose the lightweight CNN with the best comprehensive performance (MobileNetV3 or EfficientNet Lite) as the fixed feature extractor. MobileNetV3 uses depthwise separable convolution and SE attention modules [24]. EfficientNet Lite optimizes the network structure through composite scaling [23].

Compared with ResNet50, the selected lightweight backbone network reduces parameters by >95% and computation by >90% [9].

3.2.2. SVM/Random Forest Classifier Compression

For SVM classifiers, we use three compression techniques: (1) Support Vector Trimming: removing redundant support vectors to reduce counts by 30% -50% [20]; (2) Parameter quantization: Quantify weights from 32-bit floating-point numbers to 8-bit integers, reducing volume by 75% [20]; (3) Random Fourier Feature Approximation: Using random Fourier features to approximate Gaussian kernels, linearize calculations, and increase inference speed by over 80% [20].

For random forests, we use: (1) decision tree pruning: pruning low importance branches [14]; (2) Feature subsampling: Reduce input features [14].

3.2.3. Edge Inference Optimization

Convert lightweight CNN to ONNX and then to TFLite (Android) or TensorRT (Raspberry Pi) format. Use hardware acceleration (GPU, NPU, DSP). Deploy the compressed classifier as a lightweight script for real-time decision-making [10,11].

3.3. Crack Severity Classification Module

This core innovation achieves severity classification without pixel level segmentation, thereby avoiding high annotation costs.

3.3.1. Severity-Labeled Dataset Construction

According to GB 50204-2015, cracks are classified into two severity levels: minor cracks (width < 0.1mm) and obvious cracks (width \geq 0.1mm) [10]. The dataset includes 500 minor cracks and 500 obvious cracks Images from architectural scenes, marked by professional inspectors, with crack widths measured by high-precision gauges (precision 0.01mm).

3.3.2. High-Dimensional Feature Visualization

We use the t-SNE algorithm for dimensionality reduction

and visualization of whitened deep features [19]. Visualization shows that slight and obvious cracks form two distinct, compact, and clearly bounded clusters, proving that CNN features implicitly encode crack morphology information.

3.3.3. SVM/Random Forest Classifier Training

Based on high-dimensional features and severity labels, we train SVM and random forest classifiers for severity classification and compare their performance.

SVM uses Gaussian kernel functions with adaptive parameters (after random Fourier feature approximation). Random forest uses 100 decision trees with a maximum depth of 20 and a feature subsampling rate of 0.8 [5,14].

4. Experiments and Results

4.1. Experimental Setup

Dataset construction: Three parts were used in the experiment: (1) Public crack dataset: SDNET2018 (56318 images) and Crack500 (500 pixel level annotated images) [3,25]; (2) Self built detection dataset: 2000 images (1000 cracks and 1000 non cracks) from 20 factories, 30 residential buildings, and 15 commercial buildings in Beijing; (3) Self built severity dataset: 1000 crack images (500 mild, 500 obvious) with precise severity labels.

Evaluation indicators: For binary detection: accuracy, precision, recall, F1 score. Severity classification: each category indicator and overall accuracy. Lightweight performance: model size, inference time on Raspberry Pi 4B FLOP.

Experimental Design: Four parts: backbone comparison, robustness enhancement validation, lightweight performance testing, severity classification comparison.

4.2. Experimental Results and Analysis

4.2.1. Backbone Feature Extraction Performance

Table 1 shows the F1-score of four CNN models in the binary detection task (test set). It can be seen from the table that ResNet50 achieves the highest F1-score (0.92), followed by VGG16 (0.90).

Among the lightweight models, MobileNetV3 achieves a higher F1-score (0.89) than EfficientNet-Lite (0.87), with a smaller model volume and fewer FLOPs. Considering the comprehensive performance of feature discrimination, model volume, and computation amount, MobileNetV3 is selected as the backbone feature extractor of the proposed framework [21-24].

Table 1. Backbone Feature Extraction Performance Comparison

CNN Model	F1-Score (Test Set)	Model Volume	FLOPs
VGG16	0.90	528MB	15.6G
ResNet50	0.92	98MB	3.8G
MobileNetV3	0.89	12MB	0.21G
EfficientNet-Lite	0.87	18MB	0.35G

4.2.2. Feature Robustness Enhancement Performance

To verify the effectiveness of the robustness enhancement module, this paper constructs an interference test set containing 300 crack images with severe interferences (100 shadow interference images, 100 stain interference images, 100 low contrast images). Table 2 shows the performance comparison of the model before and after feature

enhancement in the clean test set and the interference test set.

It can be seen from Table 2 that after feature enhancement (ZCA Whitening + Adversarial Distillation), the F1-score of the model in the clean test set increases slightly from 0.89 to 0.90, while the F1-score in the interference test set increases significantly from 0.78 to 0.84 (an increase of 6%). At the same time, the false alarm rate and missed alarm rate in the interference test set are reduced by 37.5% and 42.9%,

respectively. These results fully verify that the feature robustness enhancement module can effectively improve the

model's anti-interference ability in complex industrial and civil building scenes [8,25].

Table 2. Performance Comparison Before and After Feature Enhancement

Model	Clean Set F1	Interference Set F1	Interference Set False Alarm Rate	Interference Set Missed Alarm Rate
MobileNetV3 + SVM (Before Enhancement)	0.89	0.78	0.08	0.14
MobileNetV3 + SVM (After Enhancement)	0.90	0.84	0.05	0.08

4.2.3. Lightweight Performance Results

Table 3 shows the lightweight performance of the proposed framework (MobileNetV3 + compressed SVM) compared with the traditional ResNet50 + SVM model and the uncompressed MobileNetV3 + SVM model. The results show that the compressed proposed framework has a model volume of only 92 KB (99.2% smaller than the traditional model and 99.3% smaller than the uncompressed lightweight model), an

inference time of 0.8 ms on Raspberry Pi 4B (94.8% faster than the traditional model and 80% faster than the uncompressed lightweight model), and FLOPs of only 12 M (99.7% lower than the traditional model). At the same time, the F1-score of the proposed framework in the binary detection task is only reduced by 0.01 compared with the uncompressed model, which is negligible. These results fully meet the requirements of real-time edge deployment [9, 10,20] .

Table 3. Lightweight Performance Comparison

Model	F1-Score (Test Set)	Model Volume	Inference Time (Raspberry Pi 4B)	FLOPs
ResNet50 + SVM (Traditional)	0.92	8.2MB	15.3ms	3.8G
MobileNetV3 + SVM (Uncompressed)	0.90	12.5MB	4.0ms	0.21G
MobileNetV3 + Compressed SVM (Proposed)	0.89	92KB	0.8ms	12M

4.2.4. Crack Severity Classification Performance

Table 4 shows the performance comparison of SVM and Random Forest classifiers in the crack severity classification task (test set). It can be seen from the table that the SVM classifier achieves a higher overall accuracy (87%) than the Random Forest classifier (84%), with higher precision and recall for slight cracks. The main reason is that the SVM classifier has better generalization performance in small-sample high-dimensional feature scenarios, which is more

suitable for the crack severity classification task in this paper [4,5] .

The t-SNE visualization results show that the features of slight cracks and obvious cracks form distinct clusters, which is consistent with the high classification accuracy of the SVM classifier. These results fully verify the feasibility of the non-segmentation-based crack severity classification method proposed in this paper, which can achieve high-precision severity evaluation with only global image labels, avoiding the high cost of pixel-level annotation [17].

Table 4. Severity Classification Performance Comparison

Classifier	Overall Accuracy	Slight Crack (Pre/Rec/F1)	Obvious Crack (Pre/Rec/F1)
SVM	87%	0.88 / 0.86 / 0.87	0.86 / 0.87 / 0.86
Random Forest	84%	0.85 / 0.82 / 0.83	0.83 / 0.85 / 0.84

5. Discussion

5.1. Effectiveness of the Proposed Scheme

Experimental results fully verify scheme effectiveness. Feature robustness enhancement (ZCA Whitening + Adversarial Distillation) improves interference scene F1-score by 6%, solving scene mismatch. Lightweight deployment achieves <100 KB model volume and <1 ms inference time on Raspberry Pi, meeting edge real-time requirements. Non-segmentation severity classification achieves 87% accuracy with only global labels, solving high annotation cost problem [8,20,25] .

5.2. Advantages Over Existing Methods

Compared with existing methods, this approach has three core advantages: (1) Scene positioning: specifically designed for industrial and civil buildings; (2) Low cost: no segmentation The paradigm avoids pixel level annotations; (3)

Scalability: Lightweight design enables edge devices to operate [1,3,9].

5.3. Limitations and Future Work

Limitations include: (1) Severity classification is limited to only two levels; (2) Generalization in extreme interference scenarios needs improvement; (3) Only perform static image detection without video stream detection.

Future work will: (1) expand severity classification to three or more levels; (2) Introduction Comparative learning and multi-scale feature fusion; (3) Optimize real-time video stream detection[21,23]

6. Conclusion

This article proposes an automatic crack recognition scheme for lightweight concrete in industrial and civil buildings based on CNN+SVM. To address the issues of scene mismatch, high quantification costs, and difficulty in

edge deployment, this solution has designed three core modules: feature robustness enhancement, lightweight deployment, and non segmentation severity classification. The experimental results show that the binary detection F1 score is 0.89, the interference scene is improved by 6%, the model volume is 92 KB, the inference time on Raspberry Pi 4B is 0.8 ms, and the severity classification accuracy is 87%, fully meeting the practical engineering requirements.

This study fills the gap in specialized detection algorithms for concrete cracks in industrial and civil buildings, providing a low-cost, efficient, and deployable technical solution for intelligent building health monitoring.

References

- [1] Cha, Y. J., Choi, W., & Büyüköztürk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 32(5), 361–378. <https://doi.org/10.1111/mice.12267>.
- [2] Zhang, A., Wang, K. C. P., Li, B., Fei, Y., & Zhang, Y. (2017). Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network. *Computer-Aided Civil and Infrastructure Engineering*, 32(10), 805–815. <https://doi.org/10.1111/mice.12295>.
- [3] Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q., & Wang, S. (2019). DeepCrack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing*, 28(3), 1498–1512. <https://doi.org/10.1109/TIP.2018.2878354>.
- [4] Kumar, A., Singh, P., & Gupta, R. (2019). CNN features with SVM for crack detection under limited samples. *Journal of Computing in Civil Engineering*, 33(5), 04019028. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000792](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000792).
- [5] Park, S., Kim, J., & Lee, H. (2020). Comparative study of CNN and CNN-SVM for crack detection. *Sensors*, 20(15), 4256. <https://doi.org/10.3390/s20154256>.
- [6] Wang, Z. Y., Zhang, W., & Li, Q. (2018). Bridge crack detection method based on Faster R-CNN. *China Civil Engineering Journal*, 51(7), 91–98.
- [7] Yamaguchi, T., Nakamura, S., & Saegusa, R. (2020). CrackAttentionNet: Attention-based network for crack detection. *IEEE Access*, 8, 148425–148435. <https://doi.org/10.1109/ACCESS.2020.3015823>.
- [8] Huang, H., Chen, C., & Zhang, J. (2021). Domain adaptation for crack detection via adversarial learning. *Automation in Construction*, 128, 103789. <https://doi.org/10.1016/j.autcon.2021.103789>.
- [9] Liu, J., Yang, X., & Lau, R. W. H. (2021). Lightweight crack detection network for edge devices. *IEEE Transactions on Intelligent Transportation Systems*, 22(12), 7689–7700. <https://doi.org/10.1109/TITS.2021.3060327>.
- [10] Fan, J., Li, H., & Zhang, Q. (2021). Research on lightweight crack detection algorithm for mobile phones. *Journal of Tsinghua University (Science and Technology)*, 61(5), 456–463.
- [11] Wang, H., Chen, M., & Zhao, W. (2021). Research on crack detection system based on UAV platform. *Journal of Zhejiang University (Engineering Science)*, 55(4), 712–720.
- [12] Wu, B., Liu, Q., & Zhang, Y. (2020). Construction of concrete crack image dataset in complex environment. *Journal of South China University of Technology (Natural Science Edition)*, 48(6), 89–97.
- [13] Ke, L., Zhang, H., & Li, M. (2019). Bridge crack detection method combining CNN features and SVM. *Journal of Beijing Jiaotong University*, 43(3), 78–85.
- [14] Yang, Y., Wang, Q., & Liu, W. (2020). Concrete crack recognition based on deep features and SVM. *Journal of Chongqing University*, 43(5), 67–75.
- [15] Niu, D., Zhang, W., & Li, Q. (2021). Application of CNN-SVM fusion model in concrete crack recognition. *Journal of Xi'an University of Architecture and Technology*, 53(4), 512–520.
- [16] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*. <https://arxiv.org/abs/1409.1556>.
- [17] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T. (2014). DeCAF: A deep convolutional activation feature for generic visual recognition. *Proceedings of the 31st International Conference on Machine Learning*, 647–655.
- [18] Luo, H., Wang, J., & Zhang, H. (2022). Tunnel crack recognition based on transfer learning and SVM. *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, 50(3), 89–96.
- [19] Chen, Z., Li, M., & Wang, Q. (2023). Long-span bridge crack recognition based on deep features. *Journal of Hunan University (Natural Sciences)*, 50(2), 112–120.
- [20] Chen, Y., Wang, L., & Li, H. (2023). Model compression for CNN-SVM based crack detection. *Journal of Computing in Civil Engineering*, 37(4), 04023018. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0001067](https://doi.org/10.1061/(ASCE)CP.1943-5487.0001067).
- [21] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>.
- [22] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4510–4520. <https://doi.org/10.1109/CVPR.2018.00474>.
- [23] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *Proceedings of the 36th International Conference on Machine Learning*, 6105–6114.
- [24] Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., Adam, H. (2019). Searching for MobileNetV3. *Proceedings of the IEEE International Conference on Computer Vision*, 1314–1324. <https://doi.org/10.1109/ICCV.2019.00140>.
- [25] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28.