SRCNN-based image transmission for autonomous vehicles in limited network areas

Anindya Afina Carmelya[1], Arief Suryadi Satyawan[2,3], Galura Muhammad Suranegara[1], Mokhamamad Mirza Etnisa Haqiqi[4], Helfy Susilawati[5], Nizar Alam Hamdani[5], Puput Dani Prasetyo Adi[2]

[1]Department of Telecommunication System, Universitas Pendidikan Indonesia, Purwakarta, Indonesia
[2]Department of Electrical Engineering, Universitas Nurtanio, Bandung, Indonesia
[3]National Research and Innovation Agency, Jakarta, Indonesia
[4]Department of Engineering, Universitas Indonesia, Depok, Indonesia
[5]Department of Electrical Engineering, Universitas Garut, Garut, Indonesia

Article Info

Article history:

Received Jul 2, 2024
Revised Sep 17, 2024
Accepted Sep 30, 2024

Keywords:

Autonomous vehicles
High-resolution images
Image transmission
Network efficiency
SRCNN

ABSTRACT

High-quality images are crucial for navigation, obstacle detection, and environmental understanding, but transmitting high-resolution images over constrained networks presents significant challenges. This study introduces an image transmission system using super-resolution convolutional neural networks (SRCNN) to enhance image quality without increasing bandwidth requirements by transmitting low-resolution images and upscaling them with SRCNN. The first phase of the research involved data collection, in which information was acquired directly from an appropriate locus to produce training, validation, and testing datasets. The second, three SRCNN models (915, 935, and 955) were trained using such a training dataset. The last was an evaluation, in which model 915 showed quick learning and stable performance with initial high loss, while model 935 had rapid convergence but potential overfitting. Model 955 achieved high initial performance. Three SRCNN model configurations were tailored to the specific needs of autonomous electric vehicles operating in limited areas, such as the locus. Input image resolution ranged from 128×128 pixels to 256×256 pixels, while output resolution varied from 256×256 pixels to 512×512 pixels. These resolutions can be acceptable for efficient image transmission over IEEE 802.11ac, but on the long range (LoRa) network, it still produces some delay.

Corresponding Author:

Anindya Afina Carmelya
Department of Telecommunication System, Universitas Pendidikan Indonesia
Purwakarta, West Java, 41115, Indonesia
Email: anindyafina@upi.edu

1. INTRODUCTION

Effective image transmission has become essential with the quick development of autonomous car technology, particularly in places with spotty network coverage. High-quality images are essential for various tasks such as navigation, obstacle detection, and environment understanding [1]–[5]. However, transmitting high-resolution images over constrained networks poses significant challenges. This condition is because these images consume massive bandwidth transmission. For instance, in our development of an autonomous vehicle operated in a limited area using IEEE 802.11ac, with an 80 MHz channel, 2 spatial streams, and in ideal conditions, the theoretical maximum bandwidth is 866 Mbps. They are accounting for overhead and typical network conditions. This condition might achieve 60-70% of the theoretical maximum, or around

520-606 Mbps [6], [7]. This bandwidth can easily support multiple simultaneous HD or UHD video streams. Another option is employing the long range (LoRa) standard, which operates with bandwidths ranging from 125 to 500 kHz. The maximum data rate of LoRa is only 5% of the minimum requirement for SD video (720×480 pixels), 50 kbps [8]–[11].

Our work offers a comprehensive method based on super-resolution convolutional neural networks (SRCNN) for enhancing picture transmission in autonomous cars. This technique minimizes data transmission loads while maintaining important visual information by transmitting low-quality images and upscaling them to high resolution on the vehicle's end, especially in limited or unstable network conditions. Better coordination and decision-making are encouraged by this approach, which improves picture quality for autonomous vehicles and facilitates communication between vehicles and central operations. Our method, which use SRCNN to upscale smaller input images in contrast to earlier research [12], shows promise in a range of image transmission scenarios within the autonomous car ecosystem.

2. METHOD

We present a workable solution for efficient picture transmission in autonomous vehicles: SRCNN. It tackles the difficulties associated with sending high-resolution pictures in bandwidth-constrained settings. By integrating SRCNN, low-resolution images can be transmitted quickly and enhanced into high-resolution images necessary for precise navigation [13], [14], making our approach viable for real-world implementation. In Figure 1, we present the process of transmitting images in our research scenario. The process of improving low-resolution photos using a SRCNN, a crucial component of our approach, is depicted in the diagram. A low-resolution image is first transmitted by a transmitter. This image is subsequently transmitted to a receiver, where its low resolution is retained.

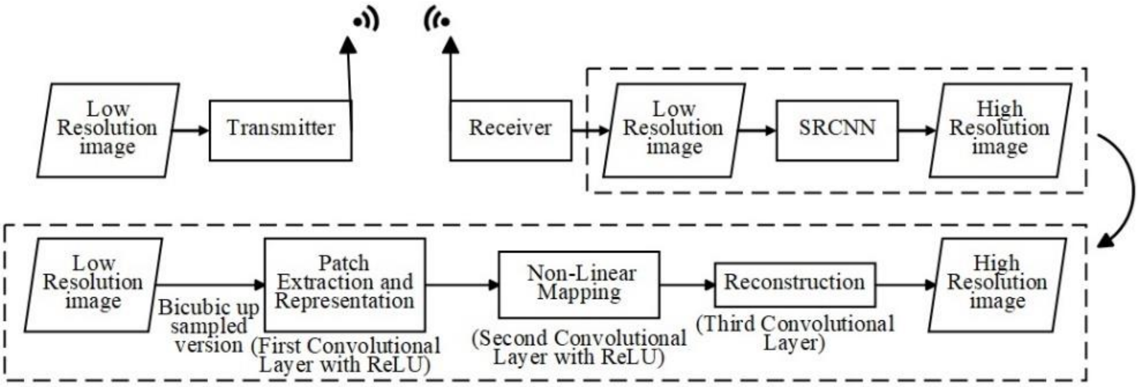

Low Resolution image
Transmitter
Receiver
Low Resolution image
SRCNN
High Resolution image
Low Resolution image
Bicubic up sampled version
Patch Extraction and Representation
(First Convolutional Layer with ReLU)
Non-Linear Mapping
(Second Convolutional Layer with ReLU)
Reconstruction
(Third Convolutional Layer)
High Resolution image

Figure 1. Super resolution-based image transmission scheme

The primary focus of the process is to enhance this low-resolution image using SRCNN to obtain a high-resolution version. The SRCNN process begins by up-sampling the low-resolution image through bicubic interpolation. This step increases the image's size to match the desired high-resolution dimensions, serving as a preliminary enhancement. Following this, the up-sampled image undergoes to patch extraction and representation. This is achieved using the first convolutional layer with rectified linear unit (ReLU) activation, which extracts small regions or patches from the image and represents them in a higher-dimensional feature space. Next, these patches are processed through a second convolutional layer with ReLU activation, performing a non-linear mapping. This layer plays an essential role because it bridges the resolution gap by figuring out the complex interactions between high-resolution and low-resolution patches. Ultimately, a third convolutional layer receives the output from this layer and uses the processed features to recreate the high-resolution image. An enormously improved high-resolution image replaces the initial low-resolution input.

A modified neural network called the SRCNN was created to enhance image resolution [15], [16]. The SRCNN has three main layers, as Table 1 illustrates. The input layer works with a low-resolution image that has been down-sampled from its original high-resolution form by an average of 2, 3, or 4. Three convolutional layers make up the SRCNN model. The first layer uses 64 filters (9×9) and ReLU activation to recover overlapping patches from the low-resolution picture and converts them into high-dimensional feature vectors. The second layer maps these vectors to another set of high-dimensional feature maps using 32 filters (1×1) with ReLU. The third layer reconstructs the high-resolution image using filters (5×5), with the number of filters depending on the image type, producing a linear output. The model is trained using a mean squared error (MSE) loss function in order to minimize the difference between the reconstructed high-resolution image and the original high-resolution image [17]. The model is then optimized using stochastic gradient descent (SGD), which makes use of data augmentation methods like rotation, scaling, and mirroring to boost robustness. The mean squared error (MSE) loss function is used by SRCNNs to increase image resolution. In order to minimize the discrepancy between the real high-resolution images and the high-resolution images predicted by SRCNN, this function is essential for training [18], [19].

Table 1. The network architecture of SRCNN

Layer	Purpose	Layer type	Filter size	Number of filters	Stride	Activation function	Output
Input	Low-resolution image	-	-	-	-	-	Low-resolution image
Layer 1	Extracts overlapping patches and maps to high-dimensional feature vectors	Convolutional layer	9×9	64	1	ReLU	Feature maps representing extracted patches
Layer 2	Maps feature vectors to another set of high-dimensional feature vectors	Convolutional layer	1×1	32	1	ReLU	New set of high-dimensional feature maps
Layer 3	Reconstructs the high-resolution image from the feature vectors	Convolutional layer	5×5	1 (grayscale) or 3 (RGB)	1	None (linear activation)	Final high-resolution image

Mathematically, the MSE loss function is defined as follows (1) [20]:

$$MSE\ Loss = \frac{1}{N}\sum_{i=1}^{N}(I_{high}(i) - \hat{I}_{high}(i))^2 \quad (1)$$

In this case, N stands for the total number of pixels in the image, I_{high} for the ground truth high-resolution image, and $\hat{I}_{high}$ for the predicted high-resolution image generated by SRCNN. The loss function calculates the average squared difference between corresponding pixels across the entire image, providing a quantitative measure of how well the predicted output matches the actual high-resolution target. Using gradient descent and backpropagation to modify parameters, the SRCNN model minimizes the MSE loss during training, lowering prediction error and enhancing image quality. SRCNN effectively generates high-fidelity super-resolution images that closely resemble the high-resolution ground-truth images by using MSE as the primary loss function. This shows significant improvements over traditional interpolation methods using CNNs and deep learning. This super-resolution application will be employed to efficiently deliver images in autonomous electric vehicles operating in confined environments. The research process was developed in three phases, as depicted in Figure 2.

The raw dataset was acquired and prepared in the first phase. As shown in Figure 3, data collection for this stage started at the KST Samaun Samadikun BRIN in the Bandung area. With a 4:3 aspect ratio and a 30 frames per second frame rate, the Insta360 ONE RS Twin Edition 4K Boost Lens camera was used to record the data.

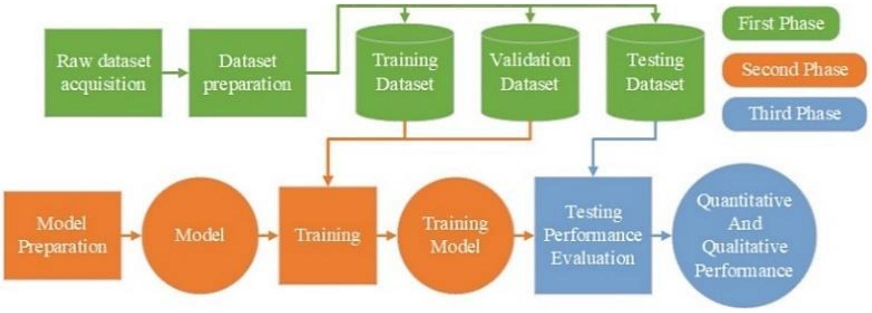


Figure 2. The research process

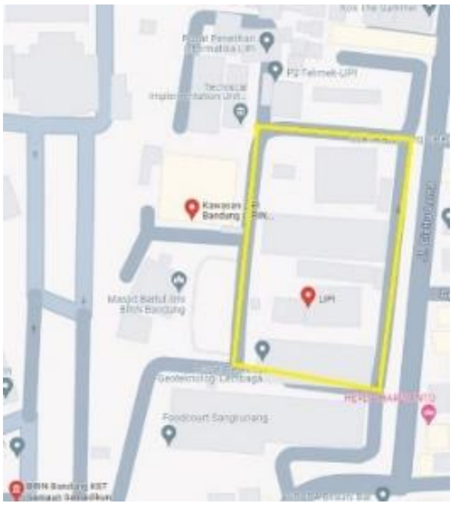

P2 Telimek-LIPI
Technical Implementation Unit...
Kawasan BRIN Bandung
Masjid Baitul Ilmi BRIN Bandung
LIPI
Jl. Sangkuriang
Foodcourt Sangkuriang
BRIN Bandung KST Samaun Samadikun

Figure 3. The location of the raw dataset collection

The second phase focuses on model preparation and training. The Adam optimiser was used with a batch size of one and a training step of 200,000. The developed SRCNN model had a layer composition with filter sizes of 9×9, 1×1 (or 3×3; 5×5) and 5×5 in its three convolutional layers. These layers correspond to the patch extraction and representation layer, the non-linear mapping layer, and the reconstruction layer. During this stage, a pre-prepared training and validation dataset was used to train the model. This phase resulted in a fully trained model, which was assessed using four important metrics at each training stage. Throughout the training procedure, training peak signal to noise ratio (PSNR), validation PSNR, training loss, and validation PSNR were tracked. After that, the model was ready for testing in the subsequent stage.

Using the previously created test dataset, the trained model's performance is assessed in the last phase. Based on the images in the testing dataset, the testing method produces the mean PSNR value, a qualitative indicator of the model's performance. To assess the model's performance quantitatively, a demonstration is carried out to highlight the caliber of the super-resolution photographs produced by the trained model. This phase makes sure that the model's efficacy is thoroughly assessed before it is put into use.

3. RESULTS AND DISCUSSION

This section will discuss comprehensively the results given from the experiment based on the three phase described in method section.

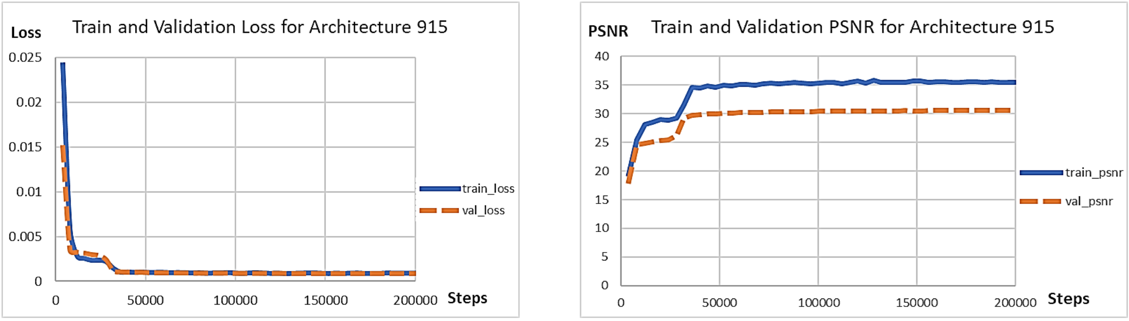
3.1. First phase results

The data collection process involved capturing videos at fifteen distinct locations within the designated area, resulting in fifteen separate videos. Each video was then converted into a series of discrete frames using Python, generating a substantial number of frames. Frames containing comparable data were eliminated to guarantee a diversified dataset. Next, three subsets of the remaining frames were created: training, validation, and testing sets [21]-[26]. From the selected frames, 322 images were categorized into eight size categories, with dimensions of 150×150, 200×200, 250×250, 300×300, 350×350, 400×400, 450×450, and 500×500 pixels, forming the training dataset. The validation dataset comprised 141 images, chosen to cover a broader range of sizes than the training dataset. The testing dataset included images scaled in three categories: ×2, ×3, and ×4. Each scale had a corresponding category for data and labels, each containing 50 images of the same dataset but in different sizes. The labels featured ground truth images with dimensions of 512×512 pixels for scales ×2 and ×4 and 510×510 pixels for scale ×3. The data category included images resized to half, a third, or a quarter of the ground truth size, depending on the scale. For testing input, the image resolutions were 256×256 pixels for scale ×2, 170×170 pixels for scale ×3, and 128×128 pixels for scale ×4. Upon completion of this phase, the training, validation, and testing datasets were prepared for subsequent phases of the research.

3.2. Second phase results

Figure 4 describes the results of training and validation for architecture 915 for both loss and PSNR. The graph in Figure 4(a) shows the training and validation loss for a model with the architecture model 915 over a series of training steps ranging from 0 to 200,000. Both the training and validation losses start relatively high, around 0.025, but show a steep decline in the early steps, decreasing significantly by around 50,000 steps. After this rapid decrease, the losses stabilise and remain low and close to each other for the remaining training steps. The convergence of both the training and validation losses to similar low values suggests that the model is performing effectively, with a strong generalisation capability, as indicated by the validation loss remaining close to the training loss and showing no signs of overfitting. The strongest aspect of this model is its ability to learn quickly and maintain low loss values, demonstrating robust performance. However, the weakness lies in the high initial loss, suggesting that further optimisation or better initialisation strategies could potentially improve the early performance of the model. Overall, the model shows excellent performance and generalisation with the given architecture.

The graph in Figure 4(b) shows the PSNR for both the training and validation data sets using the architecture model 915 over a series of training steps ranging from 0 to 200,000. Initially, both the training and validation PSNR values start relatively low, around 10 to 20 dB. As training progresses, there is a sharp increase in the PSNR values within the first 50,000 steps, with the training PSNR rising rapidly to around 35 and the validation PSNR reaching around 30. After this initial surge, both the training and validation PSNR values stabilise and remain at these levels for the remainder of the training steps. The model shows a strong initial learning phase, as evidenced by the rapid increase in PSNR, indicating that it quickly improves the quality of the reconstructed images. In addition, the stabilisation of both PSNR values suggests consistent performance over extended training. However, the model starts with relatively low PSNR values, suggesting poor initial performance that could be improved with better initialisation or pre-training strategies. Additionally, there appears to be a minor overfitting, with the model doing better on the training data than the validation data, as evidenced by the substantial difference between the training PSNR (about 35) and the validation PSNR (around 30). Overall, the model shows strong performance with rapid initial improvements and stable quality maintenance over time, although there is room for improvement in initial performance and generalisation to unseen data.



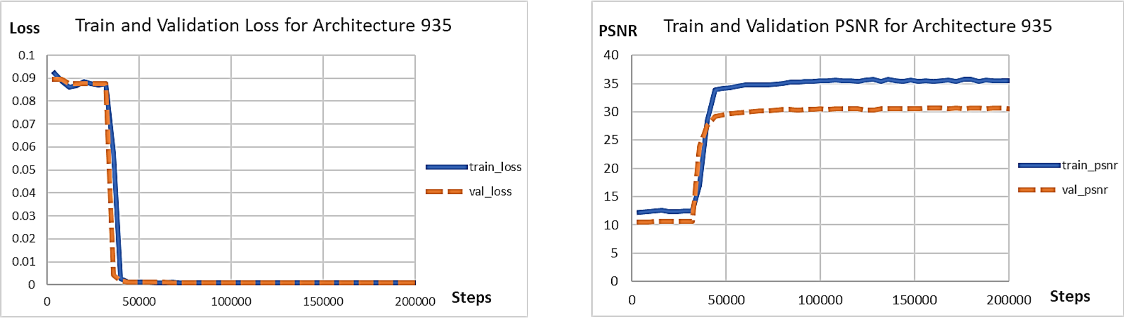
(a) (b)

Figure 4. The result of training and validation for architecture 915 (a) loss (b) PSNR

Figure 5 describes the results of training and validation for architecture 935 for both loss and PSNR. The graph in Figure 5(a) shows the training and validation loss for a model with the architecture model 935 over a series of training steps ranging from 0 to 200,000. Initially, both the training and validation losses start high, around 0.09. There is a sharp drop in loss within the first 50,000 steps, where both training and validation losses drop to around 0.01. After this sharp decline, the losses stabilize and remain consistently low for the remainder of the training steps. The model shows strengths in rapid convergence, effectively reducing losses within the first 50,000 steps, and maintaining low and stable loss values throughout the training period, indicating strong performance and no degradation over time. However, the model starts with relatively high loss values, suggesting poor initial performance that could be improved with better initialization techniques. In addition, the almost identical nature of the training and validation losses could indicate that the validation data is not sufficiently challenging or diverse compared to the training data, potentially masking overfitting. Overall, the model shows robust performance but could benefit from improvements in initial performance and validation data diversity.

The PSNR for the training and validation sets of the 935-model architecture is presented on the graph in Figure 5(b). The steps, which span from 0 to 200,000, are represented by the x-axis, while the PSNR values are displayed on the y-axis. The training PSNR is shown by the solid blue line, while the validation PSNR is shown by the dashed orange line. The model shows a rapid improvement in PSNR within the first

50,000 steps, indicating effective initial learning. The training PSNR stabilizes around 35 dB, which is relatively high and indicates that the model fits the training data well. The significant difference between the PSNRs for training and validation points to the possibility of overfitting, in which the model performs well on training data but poorly on validation data. Overall, while the model learns the training data effectively, its generalization ability needs improvement, potentially requiring techniques such as regularization, dropout or more diverse training data to improve performance on the validation set.

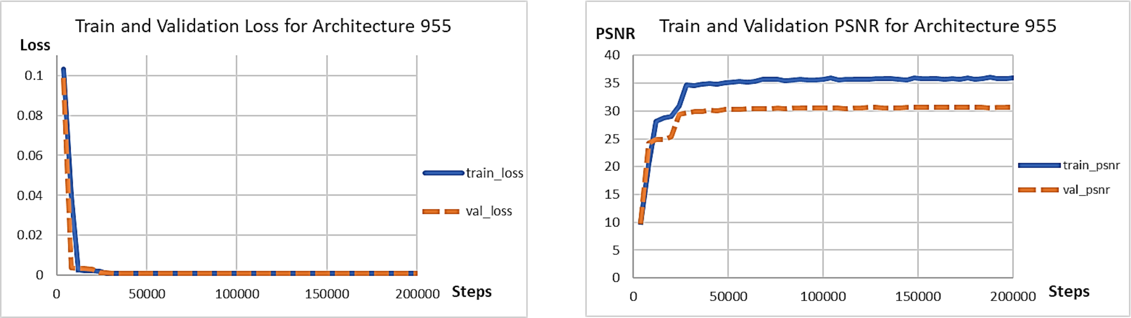


(a) (b)

Figure 5. The result of training and validation for architecture 935 (a) loss (b) PSNR

Figure 6 describes the results of training and validation for architecture 955 for both loss and PSNR. The graph in Figure 6(a) shows the loss over training steps for both the training and validation sets of model architecture 955, with the x-axis representing steps from 0 to 200,000 and the y-axis showing the loss values. Both the validation loss (dashed orange line) and the training loss (solid blue line) begin at roughly 0.1 and drop off quickly throughout the first 10,000 steps before stabilizing at close to zero for the remaining steps. The model demonstrates efficient learning with rapid loss reduction and achieves a low final loss, suggesting a good fit to the data without significant overfitting. However, the near-zero loss raises concerns about potential overfitting, as the model may be highly tuned to the training data. Furthermore, the fact that the validation loss closely follows the training loss without much variation suggests a lack of diversity in the validation set or potential problems with the validation process. Therefore, while the model shows strong initial performance, further validation is required to ensure robust generalization.

The graph in Figure 6(b) shows the PSNR over training steps for both the training and validation sets of model architecture 955, with the x-axis representing steps from 0 to 200,000 and the y-axis showing PSNR values. The solid blue line (training PSNR) and the dashed orange line (validation PSNR) both increase rapidly within the first 50,000 steps. The training PSNR stabilizes at around 35, while the validation PSNR levels off at around 30. The model shows efficient initial learning and achieves high PSNR values for both training and validation sets, indicating good performance. The model does better on the training data than the validation data, though, and a noticeable difference in the PSNRs between the training and validation sets the possibility of overfitting. Furthermore, the validation PSNR plateaus and falls short of the training PSNR, suggesting that the model's ability to generalize upon new data needs improvement.



(a) (b)

Figure 6. The result of training and validation for architecture 935 (a) loss (b) PSNR

3.3. Third phase results

Table 2 shows a comparative analysis of three SRCNN models, 915, 935, and 955, across three scale metrics: ×2, ×3 and ×4. Each metric contains the maximum, minimum and average values of PSNR for these models. Table 2 demonstrate SRCNN model 915 notable strengths, particularly with a high maximum value of 35.705 for the ×2 scale, indicating robust performance in this specific aspect. The model also maintains relatively high average values across all scales (×2: 33.549, ×3: 30.913, ×4: 29.594), demonstrating consistent performance. However, the model shows weaknesses with lower minimum values for ×3 (27.721) and ×4 (26.379), indicating variability and inconsistency in performance for these scales. The SRCNN model 935 a commendable maximum value of 35.550 for the ×2 scales, slightly lower than model 915 but still strong. It remains consistent with respectable averages (×2: 33.483, ×3: 30.926, ×4: 29.589). Nevertheless, the model's minimum values for ×2 (29.941) and ×4 (26.369) are the lowest of the three scales, indicating potential for improvement in maintaining a higher baseline performance. The SRCNN model 955 stands out with the highest maximum value for the ×2 metric (35.761), the best of all three models. It also has the highest minimum value for the ×2 scales (30.092), demonstrating strong basic performance. The average values for ×2 (33.655) and ×4 (29.596) are the highest of the three models, indicating overall robust performance. However, the ×3 metric shows a slightly lower maximum value (32.895) compared to the other models, which could be an area for improvement.

Table 2. The PSNR of test in scale ×2, ×3, ×4 for all architecture model

The PSNR of test in scale ×2, ×3, ×4 for architecture model 915			
	×2	×3	×4
Max	35.705	32.918	31.609
Min	29.959	27.721	26.379
Average	33.549	30.913	29.594
The PSNR of test in scale ×2, ×3, ×4 for architecture model 935			
	×2	×3	×4
Max	35.550	32.916	31.584
Min	29.941	27.741	26.369
Average	33.483	30.926	29.589
The PSNR of test in scale ×2, ×3, ×4 for architecture model 955			
	×2	×3	×4
Max	35.761	32.895	31.575
Min	30.092	27.723	26.374
Average	33.655	30.916	29.596

The model 955 outperforms the other two SRCNN models in terms of performance consistency on all scales. Model 935 has the lowest minimum values but has a consistent average performance, whereas model 915 comes in second with great results but significant variability. To improve their reliability, models 915 and 935 could focus on improving their minimum performance values. Overall, model 955 emerges as the most reliable and effective of the three, showing superior performance and consistency.

Examples of input and output images utilized during the testing process are shown in Figure 7. The input image for every model input that has the highest PSNR value is displayed in Figure 7(a). In the meantime, the 955-architecture model's output image, which was found to be the top performance at a scale factor of 2, is displayed in Figure 7(b). Owing to the manuscript's page limit restrictions, the resolution of the input and output photographs has been reduced by four times. This output image shows how the model can reconstruct or improve images at twice the original size without sacrificing quality. The capacity of the 955 architectures to produce crisp, detailed images highlights the remarkable performance of the system and points out the effectiveness of the algorithms and design in handling image upscaling jobs.

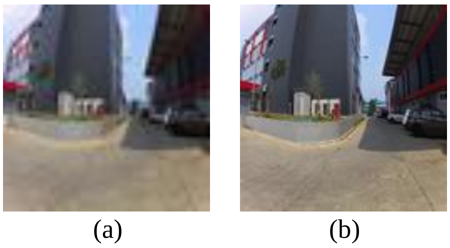

(a) (b)

Figure 7. Testing process (a) a sample input image and (b) a sample output given by super resolution (SRCNN-955)

4. CONCLUSION

This study uses SRCNN to increase the picture transmission efficiency in autonomous cars navigating limited network conditions. Three different SRCNN model configurations were created in order to meet the unique needs of autonomous electric vehicles that operate in restricted spaces, such Bandung, Indonesia's KST Samaun Samadikun. The validity of this research is supported by the use of a large dataset consisting of 765 photos, which guaranteed that the models were assessed under different conditions to achieve optimal performance.

The input image resolution in this experiment varied from 128×128 pixels to 256×256 pixels. In contrast, the scale factor used affected the output resolution, which ranged from 256×256 pixels to 512×512 pixels. Even with the smallest input image size of 128×128 pixels, the SRCNN setup managed to achieve a PSNR of 29.596 by upsizing the image to 512×512 pixels. The same model could upscale an image with an input resolution of 170×170 pixels to 510×510 pixels with a PSNR of 30.926. The final photos looked fantastic. It will be more effective to transfer these images over the IEEE 802.11ac Wi-Fi network, but there will be noticeable delays when using the LoRa network a 128×128 image will take 5.6 seconds to broadcast. This emphasizes the necessity for additional study to enhance SRCNN's functionality, especially with regard to transmission efficiency across various networks.

ACKNOWLEDGEMENTS

Special recognition is given to the Indonesia Endowment Fund for Education (*Lembaga Pengelola Dana Pendidikan*), which provided funding for this study as part of the National Research and Innovation Agency's *Riset dan Inovasi untuk Indonesia Maju* (RIIM) program (BRIN).

REFERENCES

[1] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, “Deep learning for visual understanding: a review,” *Neurocomputing*, vol. 187, pp. 27–48, Apr. 2016, doi: 10.1016/j.neucom.2015.09.116.

[2] F. Fraundorfer and D. Scaramuzza, “Visual odometry: part II: matching, robustness, optimization, and applications,” *IEEE Robotics and Automation Magazine*, vol. 19, no. 2, pp. 78–90, Jun. 2012, doi: 10.1109/MRA.2012.2182810.

[3] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

[4] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2004, doi: 10.1017/CBO9780511811685.

[5] R. Szeliski, *Computer vision*. Cham: Springer International Publishing, 2022, doi: 10.1007/978-3-030-34372-9.

[6] M. Gast, *802.11ac: A Survival Guide*. O’Reilly Media, Inc, 2013.

[7] N. S. Ravindranath, I. Singh, A. Prasad, and V. S. Rao, “Performance evaluation of IEEE 802.11ac and 802.11n using NS3,” *Indian Journal of Science and Technology*, vol. 9, no. 26, Jul. 2016, doi: 10.17485/ijst/2016/v9i26/93565.

[8] V. D. Pham, V. Vishnevsky, D. C. Nguyen, and R. Kirichek, “LoRa mesh network for image transmission: an experimental study,” *In International Conference on Next Generation Wired/Wireless Networking,* 2023, pp. 606–617, doi: 10.1007/978-3-031-30258-9_54.

[9] B. S. Chaudhari and M. Zennaro, *LPWAN technologies for IoT and M2M applications*. Elsevier, 2020, doi: 10.1016/C2018-0-04787-8.

[10] W. Guibene, J. Nowack, N. Chalikias, K. Fitzgibbon, M. Kelly, and D. Prendergast, “Evaluation of LPWAN technologies for smart cities: river monitoring use-case,” in *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, IEEE, Mar. 2017, pp. 1–5, doi: 10.1109/WCNCW.2017.7919089.

[11] D. Magrin, M. Centenaro, and L. Vangelista, “Performance evaluation of LoRa networks in a smart city scenario,” in *2017 IEEE International Conference on Communications (ICC),* IEEE, May 2017, pp. 1–7, doi: 10.1109/ICC.2017.7996384.

[12] C.-C. Wei, S.-T. Chen, and P.-Y. Su, “Image transmission using LoRa technology with various spreading factors,” in *2019 2nd World Symposium on Communication Engineering (WSCE)*, IEEE, Dec. 2019, pp. 48–52, doi: 10.1109/WSCE49000.2019.9041044.

[13] C. Dong, C. C. Loy, and X. Tang, “Accelerating the super-resolution convolutional neural network,” *In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands,* 2016, pp. 391–407, doi: 10.1007/978-3-319-46475-6_25.

[14] C. Dong, C. C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” *In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands,* 2014, pp. 184–199, doi: 10.1007/978-3-319-10593-2_13.

[15] K. Umehara, J. Ota, and T. Ishida, “Application of super-resolution convolutional neural network for enhancing image resolution in chest CT,” *Journal Digit Imaging,* vol. 31, no. 4, pp. 441–450, Aug. 2018, doi: 10.1007/s10278-017-0033-z.

[16] X. Ji, Y. Lu, and L. Guo, “Image super-resolution with deep convolutional neural network,” in *2016 IEEE First International Conference on Data Science in Cyberspace (DSC),* IEEE, Jun. 2016, pp. 626–630. doi: 10.1109/DSC.2016.104.

[17] M. Zhao, H. Peng, L. Li, and Y. Ren, “Graph attention network and informer for multivariate time series anomaly detection,” *Sensors,* vol. 24, no. 5, p. 1522, Feb. 2024, doi: 10.3390/s24051522.

[18] R. Shanmugamani, *Deep learning for computer vision: expert techniques to train advanced neural networks using TensorFlow and Keras,* Packt Publishing Ltd, 2018.

[19] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, “Enhanced deep residual networks for single image super-resolution,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* IEEE, Jul. 2017, pp. 1132–1140, doi: 10.1109/CVPRW.2017.151.

[20] A. S. Satyawan, S. U. Prini, S. A. R. Abu-Bakar, and Y. N. Wijayanto, “An elastic frame rate up-conversion for sequential omnidirectional images,” *International Journal of Advanced Science, Engineering and Information Technology*, vol. 12, no. 1, p. 158, Jan. 2022, doi: 10.18517/ijaseit.12.1.12963.
[21] J. Heaton, “Ian goodfellow, yoshua bengio, and aaron courville: deep learning,” *Genet Program Evolvable Mach*, vol. 19, no. 1–2, pp. 305–307, Jun. 2018, doi: 10.1007/s10710-017-9314-z.
[22] C. M. Bishop, *Pattern recognition and machine learning*. Springer New York, NY, 2006.
[23] H. Li, J. Li, X. Guan, B. Liang, Y. Lai, and X. Luo, “Research on overfitting of deep learning,” in *2019 15th International Conference on Computational Intelligence and Security (CIS)*, IEEE, Dec. 2019, pp. 78–81, doi: 10.1109/CIS.2019.00025.
[24] A. Géron, *Hands-on machine learning with scikit-learn, keras, and TensorFlow*, 2nd ed. O’Reilly Media, Inc, 2019.
[25] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
[26] K. P. Murphy, *Machine learning: a probabilistic perspective*, The MIT Press, 2012.

BIOGRAPHIES OF AUTHORS

Anindya Afina Carmelya was born in Bandung, Indonesia, in 2003. She is now a final-year bachelor's student in the Telecommunication System Department at Universitas Pendidikan Indonesia. For the past four months, she has started an apprenticeship in BRIN's research and development of object detection for autonomous vehicles. His current research interests are machine learning, artificial intelligence, and super resolution. She can be contacted at email: anindyafina@upi.edu.

Arief Suryadi Satyawan received his master's degree in Electrical Engineering from Institut Teknologi Bandung in Indonesia in 2007. In 2019, he received his Computer and Communication Engineering Doctorate from Waseda University, Japan. Since 1997, he has worked for the Indonesian Institute of Sciences, whose name was changed to the National Research and Innovation Agency, Indonesia, in 2021. His research interests are applications of machine learning, artificial intelligence, LIDAR-based object detection, and visual communication based on machine learning. He can be contacted at email: arie021@brin.go.id.

Galura Muhammad Suranegara holds a Master's degree in Electrical Engineering specializing in Telematics (Telecommunications and Informatics) and Telecommunication Networks from Institut Teknologi Bandung in Indonesia. Currently, he serves as the Head of the Study Program at Universitas Pendidikan Indonesia. His academic and research endeavors are centered around the fields of telecommunication and soft computing, with a particular focus on networking technologies and artificial intelligence. He can be contacted at email: galurams@upi.edu.

Mokhamamad Mirza Etnisa Haqiqi earned a Bachelor of Science in Electrical Engineering from Universitas Garut in 2017 and a Master of Science in Engineering from Universitas Indonesia in 2022. He is a lecturer in the engineering department of Universitas Garut. His research interests are on super resolution, artificial intelligence, and machine learning. He can be contacted at email: mokhammad.mirza21@ui.ac.id.

Helfy Susilawati obtained a master's degree in electrical engineering from Institut Teknologi Bandung in Bandung, Indonesia in 2014 after earning a bachelor's degree in physical education from UIN Sunan Gunung Djati Bandung, Bandung, Indonesia, in 2011. Since 2012, she has been employed as a lecturer at Universitas Garut in Indonesia. Artificial intelligence, image processing, and control systems are among her areas of interest in study. She can be contacted at email: helfy.susilawati@uniga.ac.id.

Nizar Alam Hamdani received a doctorate in Management and Education at the Indonesian University of Education, completing a master's Program at the Bandung Institute of Technology and Telkom University. He has published research in management, information technology, and educational technology. He has various experiences as a consultant in economics and information technology and as an entrepreneur. He can be contacted at email: nizar_hamdani@uniga.ac.id.

Puput Dani Prasetyo Adi graduated with honors from STMIK AKAKOM Yogyakarta in 2008 with a bachelor's degree in informatic engineering, Universitas Hasannudin in Makassar, Indonesia in 2011 with a master's degree in engineering, and Kanazawa University in Japan in 2020 with a doctorate. His primary areas of study include low power wide area (LPWA) and low power wide area networks (LPWAN) employing various RF technologies and long range (LoRa) radio frequency applications. Since 2022, he has worked in the Telecommunications Research Center for the National Research and Innovation Agency (BRIN). He can be contacted at email: pupu008@brin.go.id.